2nd ed

SQT

Ground Operations Aerospace Language

(NASA-CR-136778) GROUND OPERATIONS AEROSPACE LANGUAGE (GOAL). VOLUME 4: INTERPRETIVE CODE TRANSLATOR Final Report (International Business Machines Corp.) ~~149~~ 150 p HC $9.50 CSCL 09B G3/08

N74-15890

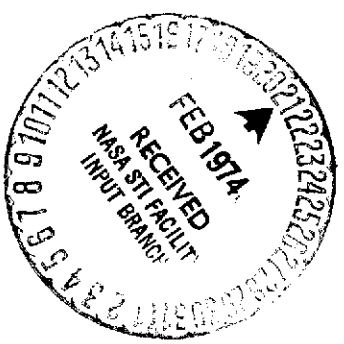
Unclas 28968

GOAL

Final Report

Volume IV

Interpretive Code Translator



31 July 1973

Ground Operations Aerospace Language

GOAL

Final Report

Volume IV

Interpretive Code Translator

Contract NAS 10-6900

Approved by: J. W. Handley

J. W. Handley, Manager
Systems Programming and
Advanced Programs

IBM Federal Systems Division

31 July 1973

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont)

1.0 INTRODUCTION

This specification identifies and describes the principal functions and elements of the Interpretive Code Translator which has been developed for use with the GOAL Compiler.

This translator enables the user to convert a compiled GOAL program to a highly general binary format which is designed to enable interpretive execution.

The translator program provides user controls which are designed to enable the selection of various output types and formats.

These controls provide a means for accommodating many of the implementation options which are discussed in the Interpretive Code Guideline document.

The technical design approach is given in Section 2.0. The relationship between the Translator and the GOAL compiler is explained and the principal functions performed by the Translator are described. Specific constraints regarding the use of the Translator are discussed.

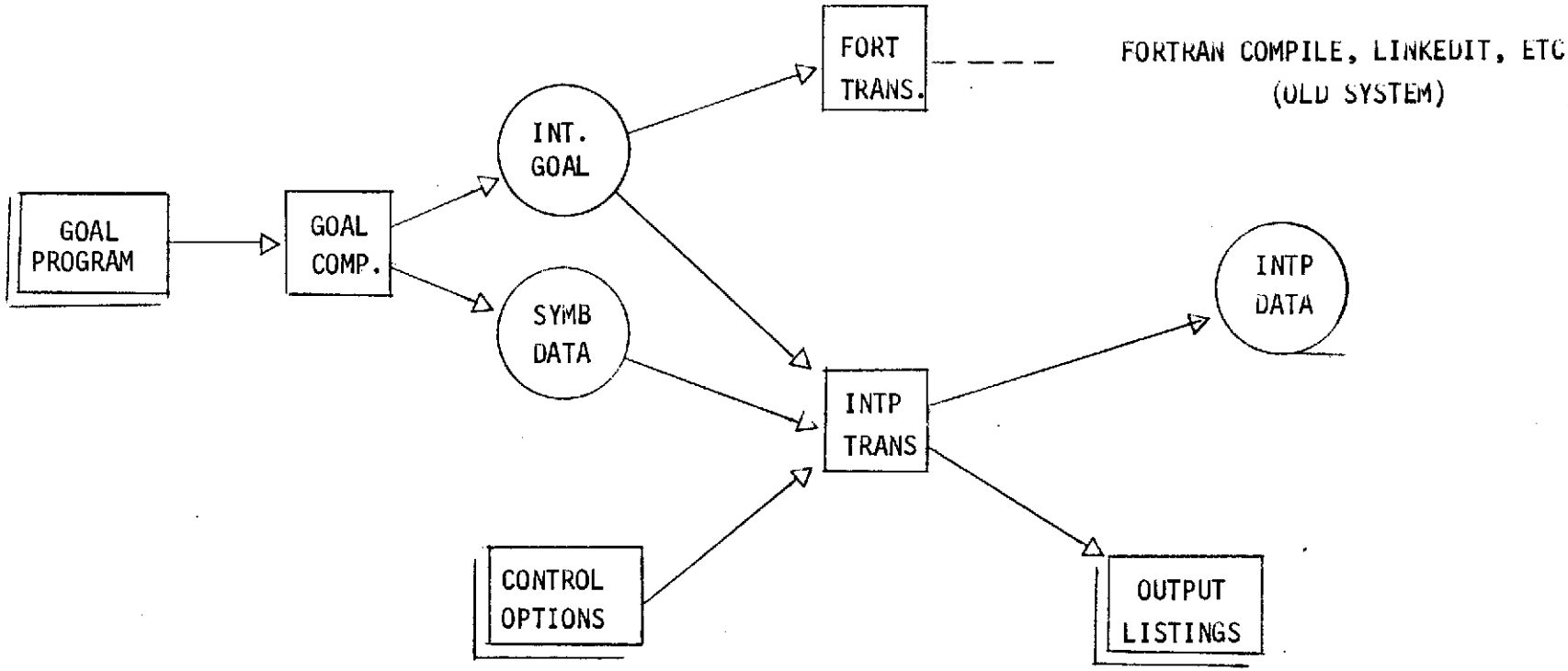
The control options are described in Section 3.0. These options enable the user to select outputs to be generated by the Translator and to control various aspects of the translation processing.

The outputs generated by the Translator are described in Section 4.0. These include run listings and data files.

The interpretive operators are identified in Section 5.0. These operators are defined with respect to the GOAL statements or statement elements from which they are derived. A principal function of the real time executive (interpretor/executor) will be the support of these operators.

The format of the interpretive data is specified in Section 6.0. The logical organization of the interpretive program structure is described.

The Translator processing techniques are described in Section 7.0. This discussion includes the processing flow, inputs and outputs, and general program structure. The program modules of the Translator program are identified and described.



GOAL INTERPRETIVE CODE TRANSLATOR

FIGURE I

2.0 DESIGN APPROACH

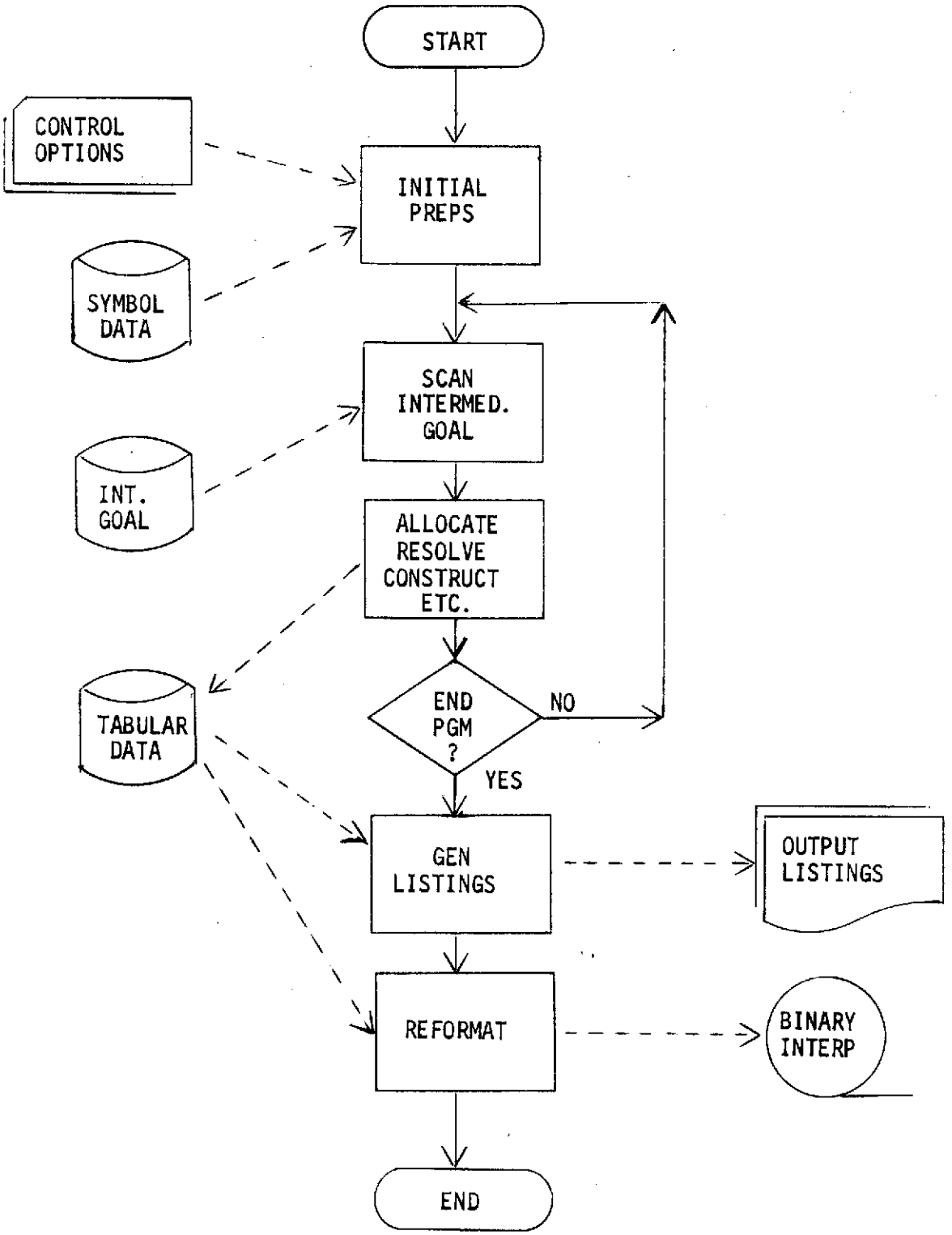
The relationship between the Interpretive Code Translator and the previous GOAL Compiler/FORTRAN Translator is shown in Figure I. Note that the Interpretive Code Translator differs from the FORTRAN translator in the following respects:

- Symbol table data not used by the FORTRAN Translator is used as input.
- The Interpretive Translator provides user control options.
- The output of the Interpretive Translator does not require a second stage of compilation to resolve internal address references.

Operational procedures will be provided to support the use of the Interpretive translator in a manner similar to the 'GOALCT' and 'GOALT' modes which are described in the GOAL Compiler Report (Volume II).

The principal processing functions performed during translation are shown on Figure II. Following is a brief description of those functions. Specific details are given in other sections of this document.

- Initial Preps - General program initialization is performed to prepare for the run. User control cards are read and the program logic is adjusted to accommodate the selected options. The 'symbol table' file is loaded to provide access to information regarding symbolic references contained in the original GOAL program.
- Scan Intermediate GOAL file - The Intermediate GOAL file created by the compiler is scanned. Each original GOAL statement generates one or more records in this file. As each record is processed the following types of functions are performed:
  - Storage Allocation - Data space is identified and reserved for the internal data that is defined either explicitly or implicitly in the user's GOAL program.
  - Address Resolution - Relative addresses are assigned to internal data, I/O control words, and branch points.
  - Operator Construction - Operation Code/parameter sequences are generated according to the statement types contained in the original GOAL program.



GOAL INTERPRETIVE TRANSLATOR
TOP LEVEL FLOW

FIGURE II

The scan loop continues until the 'END PROGRAM' statement has been processed. The Tabular data file is generated and passed on to subsequent translation steps. This file contains interpretive data which has been 'edited' according to options specified by the user's control cards. The data is in a numeric form suitable for reformatting to generate the 'binary' interpretive output.

- Generate output listings - The output listings include storage maps, symbolic references, tabular data, and other information which may assist the user in correlating the binary output of the translator with the original source program.

- Reformat for binary interpretive output - The data file is reformatted to produce binary interpretive data according to user specification of word size, character set, blocking, etc. This data is output as a sequential file in 7 or 9 track tape format.

A list of Translator program user control options is given in Figure III. These are described in greater detail in Section 3.0.

A list of Translator program outputs is given in Figure IV. The outputs are described in greater detail in Section 4.0.

The design of the interpretive code Translator reformatting procedure presents several substantial technical problems. These arise mainly in the following areas:

- Target machine storage data format

- Address resolution/non-linear mapping

- Output media compatibility.

The following ground rules have been adopted to reduce the impact of these problem areas on system implementation:

- All integer data is handled in two's complement format.

- Floating point representation is integer/exponent format.

- Conversion control card may be provided to cover the EBCDIC character set. Default is ASCII character conversion.

- Characters are packed left justified. Unused positions contain zero bits.

- Logical bit patterns are treated as right justified integer numbers.

- Target machine word size (or multiple) must accommodate relative addresses of about four decimal digits.

- Reformatting is done to word boundary. A tabular relative address is converted to a relative word position in the output interpretive data (no split words).

- The output file contains characters/bits in the same order as those in the target program. Non-standard (tape) formats are not supported.

- The reformatting procedure requires routines written in assembler language. This is limited to those functions which cannot be effectively performed in FORTRAN.

3.0 USER CONTROL OPTIONS

This section describes the features of the Interpretive Code Translator which may be controlled by the user via control cards placed in the job input stream. The control options are listed in Figure III. For each of these the valid options are given along with the default which is used if the option is not specifically selected by the user. The format of option control card(s) is given in Figure V.

3.1 INTERNAL NAMES

If this option is selected, the symbolic names of internal data items used in the GOAL program are included in the output data. Associated address information is included to support the use of the symbolic name in GOAL output statements. If appropriate real time executive routines are provided, this option may enable the user to status and/or modify the values of the internal data in a GOAL program. The symbolic names are provided as a table which precedes the procedural operators of the GOAL program in the output data file. The format of this table is given in Figure XIII.

3.2 INTERNAL STATEMENT NUMBERS

If this option is selected, a table of the internal statement numbers assigned by the GOAL compiler is included in the output data. This enables on-line reference to statements of a GOAL program which have not been assigned statement labels. The format of this table is given in Figure X.

3.3 FUNCTION DESIGNATOR NAMES

If this option is selected the symbolic names of function designators used in the GOAL program are included in the output data. Associated address information is included to support the use of the symbolic names in GOAL output statements. If appropriate real time executive routines are provided, this option may enable the user to status and/or modify test/control points and hardware addresses via keyed in request. Function designator symbolic names are provided in the form of a table which precedes the procedural operators in the interpretive data. The format of this table is given in Figure XI.

3.4 INVALID STATEMENT OPERATORS

The interpretive operators used to control execution of GOAL statements are identified and described in Section 5.0. This option enables the user to identify operators which are to be considered invalid for a given translator run. If any of these operators are encountered an error message is given and no output interpretive data is generated. However, the scan is continued to validate the remainder of the operators. As a default, all operators described in section 5.0 are considered to be valid.

3.5 CHARACTER SET

This option enables the user to define the character set which is used to represent text data and symbolic names. If this option is not selected, character data is converted according to the conversion chart given in Fibure VI, (ASCII).

3.6 OUTPUT TAPE

The user may select either seven or nine track tape as the output media for the formatted interpretive data. If this option is not specified, a seven track tape is generated. In the seven track mode, binary data is written at six bits per character. In the nine track mode, binary data is written at eight bits per character.

3.7 TARGET WORD SIZE

This option enables the user to specify the number of bits per word in the reformatted interpretive data. This word size should be sufficient to contain relative addresses used as parameters of interpretive operators. Permissable values are 16, 24, and 32.

This word size may be a multiple of the least addressable word in the target machine. If this option is not specified a word size of 24 will be used.

3.8 RECORD SIZE

This option enables the user to specify the number of target words to be included in each output tape record. Note that word size, record size and tape character size must be compatible. If this option is not specified, a record size of 2000 will be used. Permissable values range from 500 through 5000.

3.9 TARGET WORDS/INTEGER

This option enables the user to specify the number of target words to be allocated to contain integer information. This should be selected to provide sufficient precision in arithmetical operations. If this option is not specified, one target word is used. A floating point value is represented as an integer along with an exponent (power of two). One target word is used for the exponent part. The size of the integer part is one target word unless the option is used. Permissable values are 1 or 2.

3.10 TARGET CHARACTERS/WORD

This option enables the user to specify the number of characters which may be stored in each target word. The characters are stored left justified. If this option is not specified 3 characters per word is assumed. Permissable values are 1, 2, 3, 4.

3.11 TARGET CHARACTER SIZE

This option enables the user to specify the number of bits in the character fields of the target word. This field must be sufficient to contain the character set. Note that word size, character size, characters per word, and character set must be compatible. If this option is not specified a value of 8 is used. Valid selections are 6, 7, and 8.

3.12 OUTPUT LISTINGS AND FILES

This option enables the user to select which output listings and files are to be generated during the translation run. If this option is not selected all outputs will be generated. The output options are listed in Figure IV and described in Section 4.0 of this document. The selection control card format is shown in Figure V.

4.0 OUTPUT OPTIONS

This section describes the outputs generated by the GOAL Interpretive Translator. These outputs are optional and may be selected in any combination using the output option control card as described in Section 3.14. The listing examples given in this section illustrate general content and organization.

4.1 PROGRAM CONTROL BLOCK

This listing provides a summary of the principal characteristics of the Interpretive GOAL Program. This information is always written as the initial record of the interpretive data output file. An example of this listing is given in Figure VII.

4.2 INTERNAL NAMES MAP

This listing contains the symbolic names of all declared internal data items used in the GOAL program. For each name the data control block is also given. This control block is described in Sections 5.0 and 6.0. An example of this listing is given in Figure VIII.

4.3 STATEMENT LABEL MAP

This listing contains all of the statement labels used in the GOAL program. For each label the relative address of the associated interpretive code block is given. An example of this listing is given in Figure IX.

4.4 INTERNAL STATEMENT NUMBER MAP

This listing contains all of the internal statement numbers assigned to the GOAL program by the compiler. These are the same as those given in the compilation expanded source listing. For each internal statement number the address of the associated interpretive code block is given. An example of this listing is given in Figure X.

4.5 FUNCTION DESIGNATOR MAP

This listing contains the symbolic names of all Function Designators used in the GOAL program. For each name the Function Designator control block is also given. This control block is described in Sections 5.0 and 6.0. An example of this listing is given in Figure XI.

4.6 PROCEDURAL OPERATOR LISTING

This listing contains all procedural operators of the interpretive GOAL program. These operators are described in Section 5.0. The format of the parameter lists assorted with each operator is given in Section 6.0. An example of this listing is given in Figure XII.

4.7 DATA ALLOCATION MAP

This listing shows the location and contents of the areas allocated to internal data for the GOAL program. An example of this listing is given in Figure XIII.

4.8 EXTERNAL REFERENCE LISTING

This listing identifies all external references contained in the GOAL program. For each external reference the address of the control block used to establish linkage is given. These blocks are described in Sections 5.0 and 6.0. An example of this listing is given in Figure XIV.

4.9 TABULAR PROGRAM FILE

The tabular program file listing is essentially a 'dump' of the interpretive GOAL output file. The address and contents of each word of the interpretive code is listed. An example of this listing is given in Figure XV.

4.10 REFORMATTED PROGRAM FILE

This file contains all of the information indicated in the tabular program file listing. However, this data has been reformatted to conform to the control options specified by the user. This file is written on a 7 or 9 track tape.

TRANSLATOR CONTROL OPTIONS

1. INTERNAL NAMES
2. INTERNAL STATEMENT NUMBERS
3. FUNCTION DESIGNATOR NAMES
4. INVALID STATEMENT OPERATORS
5. CHARACTER SET
6. 7 TRACK OR 9 TRACK
7. TARGET WORD SIZE
8. RECORD SIZE
9. TARGET WORDS/INTEGER
10. TARGET WORDS/FLOATING POINT
11. TARGET CHAR/WORD
12. TARGET CHAR SIZE

FIGURE III

TRANSLATOR OUTPUT OPTIONS

- PROGRAM CONTROL BLOCK LISTING
- INTERNAL NAMES
- STATEMENT LABELS
- INTERNAL STATEMENT LABELS
- FUNCTION DESIGNATOR NAMES
- PROCEDURAL OPERATOR LISTING
- DATA ALLOCATION
- EXTERNAL REFERENCE LISTING
- TABULAR PROGRAM FILE
- FUNCTION DESIGNATOR I/O TABLE
- INTERNAL NAMES DDCB'S

FIGURE IV

MASTER CARD

CARD COLUMN	OPTION	CONTENTS
1	-	*
6	3.1	Ø or X*
16	3.2	Ø or X*
21	3.3	Ø or X*
25	3.6	7 or 9
29-30	3.7	16, 24 or 32
32-35	3.8	500 through 5000
40	3.9	1 or 2
45	3.10	1, 2, 3 or 4
50	3.11	6, 7 or 8

Note * - X causes option to be suppressed

Option Control Card Formats

FIGURE V-a

INVALID OPERATORS - OPTION 3.5

CARD COLUMN	CONTENTS
1	Ø
4-5	nn
9-10	nn
14-15	nn
19-20	nn
etc.	. . .

CHARACTER SET - OPTION 3.6

CARD COLUMN	CONTENTS
1	C
3-5	nnn
6	char
8-10	nnn
11	char
13-15	nnn
16	char
etc.	. . .

FIGURE V-b

TRANSLATOR OUTPUTS - OPTION 3.14

CARD COLUMN	CONTENTS	OPTION
1	L	...
6	X or blank	4.1
11	X or blank	4.2
16	X or blank	4.3
21	X or blank	4.4
26	X or blank	4.5
31	X or blank	4.6
36	X or blank	4.7
41	X or blank	4.8
46	X or blank	4.9
51	X or blank	4.10

NOTE: The 'X' PUNCH SUPPRESSES THE OPTION

FIGURE V-c

TABLE OF ASCII CHARACTERS

The following are all even parity:

7 BIT CODE	DEC.	ASCII CHAR.	GEN	
000	00	NUL	ⒸⓈP	Null
001	01	SOH	ⒸA	Start of Heading
002	02	STX	ⒸB	Start of text
003	03	EXT	ⒸC	End of text
004	04	EOT	ⒸD	End of Transmission
005	05	ENQ	ⒸE	Enquiry
006	06	ACK	ⒸF	Acknowledge
007	07	BEL	ⒸG	Bell
010	08	BS	ⒸH	Backspace
011	09	HT	ⒸI	Horizontal Tabulation
012	10	LF	LF, ⒸJ	Line Feed
013	11	VT	ⒸK	Vertical Tabulation
014	12	FF	ⒸL	Form Feed
015	13	CR	CR, ⒸM	Carriage Return
016	14	SO	ⒸN	Shift out
017	15	SI	Ⓒø	Shift in
020	16	DLE	ⒸP	Data Link Escape
021	17	DC1	ⒸQ	Device control 1
022	18	DC2	ⒸR	Device control 2
023	19	DC3	ⒸS	Device control 3
024	20	DC4	ⒸT	Device control 4
025	21	NAK	ⒸU	Negative acknowledge
026	22	SYN	ⒸV	Synchronous idle
027	23	ETB	ⒸW	End of transmission block
030	24	CAN	ⒸX	Cancel
031	25	EM	ⒸY	End of medium
032	26	SUB	ⒸZ	Substitute
033	27	ESC	ESC, ⒸⓈK	Escape

FIGURE VI-a

7 BIT CODE	DEC.	ASCII CHAR.	GEN	
034	28	FS	ⒸⓈL	File Separator
035	29	GS	ⒸⓈM	Group Separator
036	30	RS	ⒸⓈN	Record Separator
037	31	US	ⒸⓈO	Unit Separator
040	32	SP	Ⓢ1	Space
041	33	!	Ⓢ1	Exclamation point
042	34	"	Ⓢ2	Quote marks
043	35	#	Ⓢ3	Pound Sign
044	36	$	Ⓢ4	Dollar Sign
045	37	%	Ⓢ5	Percent Sign
046	38	&	Ⓢ6	Plus Sign
047	39	'	Ⓢ7	Apostrophe
050	40	(	Ⓢ8	Left Bracket
051	41	)	Ⓢ9	Right Bracket
052	42	*	Ⓢ:	Asterisk
053	43	+	Ⓢ;	Plus sign
054	44	,	,	Comma
055	45	-	-	Minus sign
056	46	.	.	Period
057	47	/	/	Slash
060	48	ø	ø	Zero
061	49	1	1	One
062	50	2	2	Two
063	51	3	3	Three
064	52	4	4	Four
065	53	5	5	Five
066	54	6	6	Six
067	55	7	7	Seven
070	56	8	8	Eight
071	57	9	9	Nine
072	58	:	:	Colon
073	59	;	;	Semi-colon
074	60	<	S ,	Less than sign

FIGURE VI-b

7 BIT CODE	DEC.	ASCII CHAR.	GEN	
075	61	=	S -	Equal sign
076	62	>	S -	Greater than sign
077	63	?	S /	Question mark
100	64	@	Ⓢ P	Commercial AT
101	65	A	A	
102	66	B	B	
103	67	C	C	
104	68	D	D	
105	69	E	E	
106	70	F	F	
107	71	G	G	
110	72	H	H	
111	73	I	I	
112	74	J	J	
113	75	K	K	
114	76	L	L	
115	77	M	M	
116	78	N	N	
117	79	O	O	
120	80	P	P	
121	81	Q	Q	
122	82	R	R	
123	83	S	S	
124	84	T	T	
125	85	U	U	
126	86	V	V	
127	87	W	W	
130	88	X	X	
131	89	Y	Y	
132	90	Z	Z	

FIGURE VI-c

7 BIT CODE	DEC.	ASCII CHAR.	GEN	
133	91	[	ⓈK	Left Square Bracket
134	92	\	ⓈL	Reverse Slash
135	93	]	ⓈM	Right Square Bracket
136	94	↑	ⓈN	Upward Arrow
137	95	←	ⓈO	Leftward Arrow
177	127	RUB	RUB	Rubout

An odd parity LF, CR, ESC from ©CR, ©LF, © ESC

FIGURE VI-d

PROGRAM CONTROL BLOCK LISTING

PCB ADDRESS AND CONTENTS

```
001 CONTAINS      1 THE OPTION FLAG FOR THE INTERNAL NAMES TABLE
002 CONTAINS      1 THE OPTION FLAG FOR THE STATEMENT LABEL TABLE
003 CONTAINS      1 THE OPTION FLAG FOR THE INTERNAL STATEMENT NUMBER TABLE
004 CONTAINS      1 THE OPTION FLAG FOR THE FUNCTION DESIGNATOR NAMES TABLE
005 CONTAINS      7 SEVEN OR NINE TRACK PROGRAM TAPE
006 CONTAINS     24 THE NUMBER OF BITS PER WORD
007 CONTAINS   2000 THE NUMBER OF WORDS PER RECORD
008 CONTAINS      1 THE NUMBER OF WORDS PER INTEGER
009 CONTAINS      3 THE NUMBER OF CHARACTERS PER WORD
010 CONTAINS      8 THE NUMBER OF BITS PER CHARACTER

050 CONTAINS    101 THE ADDRESS OF THE EXTERNAL REFERENCE TABLE
051 CONTAINS      4 THE NUMBER OF ENTRIES IN THE TABLE

052 CONTAINS    109 THE ADDRESS OF THE FUNCTION DESIGNATOR I/O TABLE
053 CONTAINS     25 THE NUMBER OF ENTRIES IN THE TABLE

054 CONTAINS    159 THE ADDRESS OF THE FUNCTION DESIGNATOR NAMES TABLE
055 CONTAINS     22 THE NUMBER OF ENTRIES IN THE TABLE

056 CONTAINS    226 THE ADDRESS OF THE INTERNAL NAMES TABLE
057 CONTAINS     66 THE NUMBER OF ENTRIES IN THE TABLE

058 CONTAINS    446 THE ADDRESS OF THE DATA DEFINITION CONTROL BLOCK TABLE
059 CONTAINS     63 THE NUMBER OF ENTRIES IN THE TABLE

060 CONTAINS    878 THE ADDRESS OF THE STATEMENT LABEL TABLE
061 CONTAINS     20 THE NUMBER OF ENTRIES IN THE TABLE

062 CONTAINS    918 THE ADDRESS OF THE INTERNAL STATEMENT NUMBER TABLE
063 CONTAINS    138 THE NUMBER OF ENTRIES IN THE TABLE

064 CONTAINS   1056 THE ADDRESS OF THE DATA AREA
065 CONTAINS   1421 THE ADDRESS OF THE LAST DATA WORD IN THE FILE

066 CONTAINS   2000 THE ADDRESS OF THE FIRST PROCEDURAL OPERATOR
067 CONTAINS   3349 THE ADDRESS PLUS ONE OF THE LAST WORD IN THE FILE
```

FIGURE VII

INTERNAL NAMES TABLE LISTING

FILE ADDRESS	FILE CONTENT	DESCRIPTION/TEXT
226	3	THE NUMBER OF CHARACTERS IN THE NAME
227	B11	
228	446	THE DDCB ADDRESS FOR THE NAME
229	2	THE NUMBER OF CHARACTERS IN THE NAME
230	C1	
231	-1	THE DDCB ADDRESS FOR THE NAME
232	2	THE NUMBER OF CHARACTERS IN THE NAME
233	C2	
234	-1	THE DDCB ADDRESS FOR THE NAME
235	2	THE NUMBER OF CHARACTERS IN THE NAME
236	C3	
237	-1	THE DDCB ADDRESS FOR THE NAME
238	4	THE NUMBER OF CHARACTERS IN THE NAME
239	MON E	
241	449	THE DDCB ADDRESS FOR THE NAME
242	3	THE NUMBER OF CHARACTERS IN THE NAME
243	MQ1	
244	452	THE DDCB ADDRESS FOR THE NAME
245	3	THE NUMBER OF CHARACTERS IN THE NAME
246	M55	
247	456	THE DDCB ADDRESS FOR THE NAME

FIGURE VIII-a

DATA DEFINITION CONTROL BLOCK LISTING

FILE ADDRESS	FILE CONTENT	DESCRIPTION/TEXT
362	1	NUMERIC DATA TYPE CODE
363	972	ADDRESS OF THE DATA
364	142	ADDRESS OF THE NAME
375	9	NUMERIC TABLE TYPE CODE
376	980	TABLE DATA ADDRESS
377	164	TABLE NAME ADDRESS
378	2	NUMBER OF ROWS
379	3	NUMBER OF COLUMNS
380	145	COLUMN NAME ADDRESS
381	148	COLUMN NAME ADDRESS
382	151	COLUMN NAME ADDRESS
383	99	ROW NAME ADDRESS
384	102	ROW NAME ADDRESS
385	47	FD I/O TABLE ADDRESS
386	49	FD I/O TABLE ADDRESS
685	5	NUMERIC LIST TYPE CODE
686	1214	ADDRESS OF THE DATA
687	271	ADDRESS OF THE NAME
688	3	NUMERIC LIST LENGTH
535	11	STATE TABLE TYPE CODE
536	1097	TABLE DATA ADDRESS
537	206	TABLE NAME ADDRESS
538	2	NUMBER OF ROWS
539	3	NUMBER OF COLUMNS
540	145	COLUMN NAME ADDRESS
541	148	COLUMN NAME ADDRESS
542	151	COLUMN NAME ADDRESS
543	123	ROW NAME ADDRESS
544	126	ROW NAME ADDRESS
545	63	FD I/O TABLE ADDRESS
546	65	FD I/O TABLE ADDRESS
743	7	STATE LIST TYPE CODE
744	1279	ADDRESS OF THE DATA
745	311	ADDRESS OF THE NAME
746	3	STATE LIST LENGTH
747	3	STATE DATA TYPE CODE
748	1282	ADDRESS OF THE DATA
749	315	ADDRESS OF THE NAME

FIGURE VIII-b

368	2	QUANTITY DATA TYPE CODE
369	976	ADDRESS OF THE DATA
370	158	ADDRESS OF THE NAME
371	3	QUANTITY UNITS CODE
481	10	QUANTITY TABLE TYPE CODE
482	1065	TABLE DATA ADDRESS
483	194	TABLE NAME ADDRESS
484	1	NUMBER OF ROWS
485	0	NUMBER OF COLUMNS
486	0	NUMBER OF COLUMN NAMES
487	75	ROW NAME ADDRESS
488	31	FD I/O TABLE ADDRESS
489	31	QUANTITY UNITS CODE
712	6	QUANTITY LIST TYPE CODE
713	1250	ADDRESS OF THE DATA
714	283	ADDRESS OF THE NAME
715	3	QUANTITY LIST LENGTH
716	1	QUANTITY UNITS CODE
717	1	QUANTITY UNITS CODE
718	1	QUANTITY UNITS CODE

638	12	TEXT TABLE TYPE CODE
639	1155	TABLE DATA ADDRESS
640	236	TABLE NAME ADDRESS
641	1	NUMBER OF ROWS
642	5	NUMBER OF COLUMNS
643	8	TEXT DATA LENGTH
644	0	NUMBER OF COLUMN NAMES
645	111	ROW NAME ADDRESS
646	55	FD I/O TABLE ADDRESS
653	4	TEXT DATA TYPE CODE
654	1175	ADDRESS OF THE DATA
655	245	ADDRESS OF THE NAME
656	2	TEXT DATA LENGTH
759	8	TEXT LIST TYPE CODE
760	1286	ADDRESS OF THE DATA
761	331	ADDRESS OF THE NAME
762	10	TEXT LIST LENGTH
763	6	TEXT DATA LENGTH

FIGURE VIII-c

STATEMENT LABEL TABLE LISTING

FILE ADDRESS	FILE CONTENT	DESCRIPTION/TEXT
878	1	STEP NUMBER
879	2290	THE OPERATOR ADDRESS
880	3	STEP NUMBER
881	2331	THE OPERATOR ADDRESS
882	10	STEP NUMBER
883	2379	THE OPERATOR ADDRESS
884	20	STEP NUMBER
885	2388	THE OPERATOR ADDRESS
886	30	STEP NUMBER
887	2406	THE OPERATOR ADDRESS
888	40	STEP NUMBER
889	2412	THE OPERATOR ADDRESS
890	100	STEP NUMBER
891	2278	THE OPERATOR ADDRESS
892	300	STEP NUMBER
893	3108	THE OPERATOR ADDRESS
894	301	STEP NUMBER
895	3126	THE OPERATOR ADDRESS
896	302	STEP NUMBER
897	3142	THE OPERATOR ADDRESS
898	303	STEP NUMBER
899	3161	THE OPERATOR ADDRESS

FIGURE IX

INTERNAL STATEMENT NUMBER TABLE LISTING

FILE ADDRESS	FILE CONTENT	DESCRIPTION/TEXT
918	2000	THE OPERATOR ADDRESS FOR ISN 1
919	-1	THE OPERATOR ADDRESS FOR ISN 2
920	2003	THE OPERATOR ADDRESS FOR ISN 3
921	2003	THE OPERATOR ADDRESS FOR ISN 4
922	2003	THE OPERATOR ADDRESS FOR ISN 5
923	2003	THE OPERATOR ADDRESS FOR ISN 6
924	2003	THE OPERATOR ADDRESS FOR ISN 7
925	2003	THE OPERATOR ADDRESS FOR ISN 8
926	2003	THE OPERATOR ADDRESS FOR ISN 9
927	2003	THE OPERATOR ADDRESS FOR ISN 10
928	2003	THE OPERATOR ADDRESS FOR ISN 11
929	2003	THE OPERATOR ADDRESS FOR ISN 12
930	2003	THE OPERATOR ADDRESS FOR ISN 13
931	2003	THE OPERATOR ADDRESS FOR ISN 14
932	2003	THE OPERATOR ADDRESS FOR ISN 15
933	2003	THE OPERATOR ADDRESS FOR ISN 16
934	2003	THE OPERATOR ADDRESS FOR ISN 17
935	2003	THE OPERATOR ADDRESS FOR ISN 18

FIGURE X

FUNCTION DESIGNATOR I/O TABLE LISTING

FILE ADDRESS	FILE CONTENT	DESCRIPTION/TEXT
109	5	THE FUNCTION DESIGNATOR TYPE CODE
110	1	THE FUNCTION DESIGNATOR ADDRESS
111	5	THE FUNCTION DESIGNATOR TYPE CODE
112	3	THE FUNCTION DESIGNATOR ADDRESS
113	5	THE FUNCTION DESIGNATOR TYPE CODE
114	4	THE FUNCTION DESIGNATOR ADDRESS
115	2	THE FUNCTION DESIGNATOR TYPE CODE
116	2001	THE FUNCTION DESIGNATOR ADDRESS
117	2	THE FUNCTION DESIGNATOR TYPE CODE
118	2002	THE FUNCTION DESIGNATOR ADDRESS
119	2	THE FUNCTION DESIGNATOR TYPE CODE
120	2003	THE FUNCTION DESIGNATOR ADDRESS
121	2	THE FUNCTION DESIGNATOR TYPE CODE
122	2004	THE FUNCTION DESIGNATOR ADDRESS
123	4	THE FUNCTION DESIGNATOR TYPE CODE
124	4001	THE FUNCTION DESIGNATOR ADDRESS

FIGURE XI-a

FUNCTION DESIGNATOR NAMES TABLE LISTING

FILE ADDRESS	FILE CONTENT	DESCRIPTION/TEXT
159	3	THE NUMBER OF CHARACTERS IN THE NAME
160	LA1	
161	115	THE DDCB ADDRESS FOR THE NAME
162	3	THE NUMBER OF CHARACTERS IN THE NAME
163	LA2	
164	117	THE DDCB ADDRESS FOR THE NAME
165	3	THE NUMBER OF CHARACTERS IN THE NAME
166	LA3	
167	119	THE DDCB ADDRESS FOR THE NAME
168	3	THE NUMBER OF CHARACTERS IN THE NAME
169	LA4	
170	121	THE DDCB ADDRESS FOR THE NAME
171	3	THE NUMBER OF CHARACTERS IN THE NAME
172	LD1	
173	123	THE DDCB ADDRESS FOR THE NAME
174	3	THE NUMBER OF CHARACTERS IN THE NAME
175	LD2	
176	125	THE DDCB ADDRESS FOR THE NAME
177	3	THE NUMBER OF CHARACTERS IN THE NAME
178	LD3	
179	127	THE DDCB ADDRESS FOR THE NAME

FIGURE XI-b

PROCEDURAL OPERATOR LISTING

<OPERATOR>	ADR= 2001	LENGTH= 2	8	2		
<OPERATOR>	ADR= 2003	LENGTH= 5	2	5	992	1	-1
<OPERATOR>	ADR= 2008	LENGTH= 5	2	5	1002	1	1171
<OPERATOR>	ADR= 2013	LENGTH= 5	2	5	1006	1	-1
<OPERATOR>	ADR= 2034	LENGTH= 19	5	19	4	0	10
			1	8	1	-14	31
			33	35	37	2	-18
			-1	3	10	0	
<OPERATOR>	ADR= 2053	LENGTH= 26	5	26	2	0	13
			2	9	11	1	-15
			1	-21	31	33	2
			-19	-1	3	10	0
			2	-25	-1	3	20
			0				

FIGURE XII

DATA AREA LISTING

FILE ADDRESS	FILE CONTENT IN GROUPS OF TEN									
972	3	0	-1	0	-11	-1	-11	-1	-3	-1
982	3	-1	3	0	10	0	0	0	255	0
992	0	0	-1	0	-2	0	-3	0	-4	0
1002	0	0	0	0	0	-1	0	-2	0	-3
1012	0	-4	0	0	0	0	0	-3	-1	3
1022	-1	3	0	10	0	0	0	255	0	0
1032	0	1	0	11	-1	11	-1	-3	-1	3
1042	-1	0	0	99	0	-99	0	-275251	-18	0
1052	0	1	0	2	0	3	0	4	0	0
1062	0	0	0	0	99	0	-99	0	-275251	-18
1072	99	0	-99	0	-275251	-18	0	0	1	0
1082	2	0	3	0	4	0	0	0	0	0
1092	5	0	5	0	0	1	0	0	1	1
1102	1	0	0	1	0	0	1	0	0	0
1112	0	0	1	0	0	0	0	0	0	0
1122	1	1	1	1	1	1	0	0	0	1
1132	0	0	0	0	-473704896	-705609664	1073745174	0	0	0
1142	-473704896	-705609664	1073745174	0	-1002422208	-705609664	0	0	0	0
1152	0	0	0	0	0	0	0	0	0	0
1162	0	0	0	0	0	1077952576	1077952576	1077952576	1073758382	2
1172	0	10	0	-471384000	-471384000	-471384000	0	0	0	0
1182	0	-471384000	0	0	0	0	0	0	0	0
1192	0	0	0	0	0	0	0	0	0	0
1202	0	0	10	0	-11	-1	0	0	0	0
1212	0	0	0	0	0	0	0	0	0	0
1222	0	0	0	0	0	0	0	0	0	0
1232	0	0	0	0	0	0	0	0	-11	-1
1242	11	-1	0	0	0	0	0	0	0	0
1252	0	0	0	0	0	0	0	0	0	0
1262	0	0	0	0	0	0	0	0	0	0
1272	0	0	0	26	50	5	0	0	0	0
1282	0	0	0	0	0	0	0	0	0	0
1292	0	0	0	0	0	0	0	0	0	0
1302	0	0	0	0	14935872	-750763968	14935872	-750763968	4210752	1077952576
1312	4210752	1077952576	4210752	1077952576	0	0	0	0	0	0
1322	14935872	-739033024	1077952576	4210752	1077952576	1077952576	0	0	0	0
1332	0	0	0	0	11	0				

END OF DATA AREA LISTING

FIGURE XIII

EXTERNAL REFERENCE TABLE LISTING

FILE ADDRESS	FILE CONTENT	DESCRIPTION/TEXT
1	-1	I.D. CODE NUMBER
2	-1	ADDRESS OF THE EXTERNAL REFERENCE
3	4	THE NUMBER OF CHARACTERS IN THE NAME
4	PGM 1	
6	-1	I.D. CODE NUMBER
7	-1	ADDRESS OF THE EXTERNAL REFERENCE
8	4	THE NUMBER OF CHARACTERS IN THE NAME
9	SUB 2	
11	-1	I.D. CODE NUMBER
12	-1	ADDRESS OF THE EXTERNAL REFERENCE
13	5	THE NUMBER OF CHARACTERS IN THE NAME
14	TES TA	

FIGURE XIV

TABULAR PROGRAM FILE LISTING

PROGRAM CONTROL BLOCK

FILE ADDRESS	FILE CONTENT IN GROUPS OF TEN									
1	1	1	1	1	7	24	2000	1	3	8
11	16	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0	1
51	4	25	25	75	22	142	66	362	63	794
61	20	834	138	972	1337	2001	3361	0	0	0
71	0	0	0	0	0	0	0	0	0	0
81	0	0	0	0	0	0	0	0	0	0
91	0	0	0	0	0	0	0	0	0	0

TABULAR FILE THROUGH THE LAST DATA WORD

FILE ADDRESS	FILE CONTENT IN GROUPS OF TEN									
1	-1	-1	4	-674769856	-247447488	-1	-1	4	-488324544	-230670272
11	-1	-1	5	-473570752	-473874368	1	-1	16	-230670272	1077952576
21	1077952576	1077952576	1077952576	1077952576	5	1	5	3	5	4
31	2	2001	2	2002	2	2003	2	2004	4	4001
41	4	4002	4	4003	4	4004	1	1001	1	1002
51	1	1003	1	1004	5	5001	5	5002	5	5003
61	5	5004	3	3001	3	3002	3	3003	3	3004
71	7	7001	6	6	3	-742264512	31	3	-742264256	33
81	3	-742264000	35	3	-742263744	37	3	-742067904	39	3
91	-742067648	41	3	-742067392	43	3	-742067136	45	3	-490606272
101	47	3	-490606016	49	3	-490605760	51	3	-490605504	53
111	3	-490475200	55	3	-490474944	57	3	-490474688	59	3
121	-490474432	61	3	-490409664	63	3	-490409408	65	3	-490409152
131	67	3	-490408896	69	3	-490081984	71	5	-488324544	-252624832
141	73	3	-1024331456	362	2	-1007599552	-1	2	-1007534016	-1
151	2	-1007468480	-1	4	-724118208	-985644992	365	3	-723979968	368
161	3	-722078400	372	3	-706481856	375	3	-706481600	387	3
171	-706481344	401	3	-706481088	409	3	-706480832	423	3	-690633408
181	435	3	-673648320	438	3	-671746752	442	3	-656150208	445
191	3	-656149952	463	3	-656149696	481	3	-656149440	489	3
201	-656149184	507	3	-656148928	525	3	-488378048	535	3	-488377792
211	547	3	-488377536	561	3	-488377280	569	3	-488377024	583
221	2	-487505856	595	2	-487440320	598	3	-471600832	601	3
231	-471600576	614	3	-471600320	629	3	-471600064	638	3	-471411136
241	647	3	-470683072	650	2	-470728640	653	2	-470663104	657
251	2	-470597568	661	2	-470532032	665	2	-470466496	669	2
261	-470400960	673	4	-455748800	-247447488	677	4	-455748800	-230670272	681
271	4	-455748800	-213893056	685	5	-455747520	-739164096	689	5	-455747520
281	-739098560	703	5	-455747520	-739033024	712	4	-455747520	-247447488	719
291	4	-455747520	-230670272	723	4	-455747520	-213893056	727	4	-455747520
301	-197115840	731	5	-455744960	-739164096	735	5	-455744960	-739098560	739
311	5	-455744960	-739033024	743	4	-455744960	-247447488	747	4	-455744960
321	-230670272	750	4	-455744960	-213893056	753	4	-455744960	-197115840	756
331	5	-455744704	-739164096	759	5	-455744704	-739098560	764	5	-455744704
341	-739033024	769	5	-455744704	-738967488	774	3	-455741120	779	3
351	-455740864	782	3	-455740608	785	3	-455740352	788	3	-403574464
361	791	1	972	142	1	974	154	2	976	158
371	3	1	978	161	9	980	164	2	3	145
381	148	151	99	102	47	49	9	994	167	4
391	1	0	75	78	81	84	31	33	35	37
401	9	1006	170	1	0	0	99	47	9	1007

FIGURE XV-a

1081	0	2	0	3	0	4	0	0	0	0
1091	0	5	0	5	0	0	1	0	0	1
1101	1	1	0	0	1	0	0	1	0	0
1111	0	0	0	1	0	0	0	0	0	0
1121	0	1	1	1	1	1	1	0	0	0
1131	1	0	0	0	0	-473704896	-705609664	1073745174	0	0
1141	0	-473704896	-705609664	1073745174	0	-1002422208	-705609664	0	0	0
1151	0	0	0	0	0	0	0	0	0	0
1161	0	0	0	0	0	0	1077952576	1077952576	1077952576	1073758382
1171	2	0	10	0	-471384000	-471384000	-471384000	0	0	0
1181	0	0	-471384000	0	0	0	0	0	0	0
1191	0	0	0	0	0	0	0	0	0	0
1201	0	0	0	10	0	-11	-1	0	0	0
1211	0	0	0	0	0	0	0	0	0	0
1221	0	0	0	0	0	0	0	0	0	0
1231	0	0	0	0	0	0	0	0	0	-11
1241	-1	11	-1	0	0	0	0	0	0	0
1251	0	0	0	0	0	0	0	0	0	0
1261	0	0	0	0	0	0	0	0	0	0
1271	0	0	0	0	26	50	5	0	0	0
1281	0	0	0	0	0	0	0	0	0	0
1291	0	0	0	0	0	0	0	0	0	0
1301	0	0	0	0	0	14935872	-750763968	14935872	-750763968	4210752
1311	1077952576	4210752	1077952576	4210752	1077952576	0	0	0	0	0
1321	0	14935872	-739033024	1077952576	4210752	1077952576	1077952576	0	0	0
1331	0	0	0	0	0	11	0			

TABULAR FILE THROUGH LAST PROCEDURAL OPERATOR

FILE ADDRESS	FILE CONTENT IN GROUPS OF TEN									
2001	8	2	2	5	992	1	-1	2	5	1002
2011	1	1171	2	5	1006	1	-1	5	16	1
2021	0	10	1	8	1	-11	31	2	-15	-1
2031	3	10	0	5	19	4	0	10	1	8
2041	1	-14	31	33	35	37	2	-18	-1	3
2051	10	0	5	26	2	0	13	2	9	11
2061	1	-15	1	-21	31	33	2	-19	-1	3
2071	10	0	2	-25	-1	3	20	0	5	10
2081	1	0	10	1	8	1	368	31	5	10
2091	1	0	10	1	8	1	438	31	5	12
2101	3	0	10	1	8	1	712	31	33	35
2111	3	10	1	0	10	1	0	9	31	47
2121	3	13	1	0	13	4	0	9	31	33
2131	35	37	47	5	16	3	0	12	1	8
2141	3	445	-15	-16	31	33	35	1	1	4
2151	6	1	0	6	73	6	12	1	747	6
2161	73	1	-9	3	-12	-1	1	6	12	1
2171	747	6	73	3	547	-11	-12	1	1	6
2181	9	3	547	-9	1033	1	747	1	7	9
2191	5	1	0	9	1	779	47	7	10	7
2201	2	992	385	2	375	-10	3	7	11	1
2211	3	0	9	1	685	47	49	51	13	22
2221	9	1	1006	407	5	11	1	-15	3	445
2231	-21	-22	2	-19	-1	-1	10	0	1	1
2241	13	12	5	0	1	-7	2	-11	-1	-1
2251	10	0	13	25	9	1	0	19	5	11
2261	1	-13	1	-20	2	-17	-1	-1	10	0
2271	47	2	-24	-1	-1	30	0	35	3	0
2281	14	3	0	14	6	3	831	829	833	35
2291	3	0	12	3	0	39	27	1	0	1
2301	0	15	1	11	13	1	-16	1	-22	47
2311	2	-20	-1	3	1	0	2	-26	-1	3
2321	20	0	23	8	1	0	8	0	0	27

3001	4	-16	-1	4	-473704896	-717209536	1	13	61	6
3011	0	1	653	4	-12	-1	4	-473704896	-717209536	31
3021	14	-9	6	0	2	629	1033	4	-13	-1
3031	4	-473704896	-717209536	34	16	0	2	0	11	1
3041	9	1	-13	43	45	3	-16	-1	0	34
3051	16	0	2	0	11	1	9	1	-13	43
3061	45	3	-16	-1	1	34	15	0	1	0
3071	11	1	9	1	-12	39	3	-15	-1	1
3081	34	15	0	1	0	13	1	9	3	535
3091	-14	-15	39	1	1	33	7	0	1	0
3101	7	39	34	10	0	1	1113	567	1	9
3111	1	747	34	15	0	1	0	13	1	9
3121	3	547	-14	-15	39	1	1	36	3	0
3131	36	5	2	799	809	35	3	0	38	12
3141	1	1	-7	13	2	-11	-1	-1	10	0
3151	16	3	811	35	3	0	38	10	1	3
3161	525	-9	-10	-11	1	1	16	3	813	35
3171	3	0	40	13	7	6	8	1	747	1
3181	-10	3	-13	-1	1	16	3	815	35	3
3191	0	40	16	7	6	10	3	535	-12	1033
3201	1	-13	1	3	-16	-1	1	16	3	817
3211	35	3	0	40	16	7	6	10	3	535
3221	-12	1033	1	-13	1	3	-16	-1	1	16
3231	3	819	35	3	0	40	12	8	6	8
3241	1	747	3	535	-12	1171	1	16	3	821
3251	35	3	0	39	18	1	10	1	0	14
3261	7	12	-12	1171	1	-15	63	3	-18	-1
3271	1	16	3	823	35	3	0	39	15	1
3281	0	2	1103	545	7	10	1	-12	3	-15
3291	-1	1	36	4	1	825	35	3	0	39
3301	13	1	0	2	1103	545	8	10	2	583
3311	-13	1	16	3	827	35	3	0	41	3
3321	71	16	3	827	35	3	0	41	3	71
3331	25	5	6	0	0	42	3	0	35	3
3341	0	41	3	71	25	5	6	0	0	42
3351	3	807	37	3	0	37	3	1	15	2

END OF TABULAR FILE LISTING

FIGURE XV-c

5.0 INTERPRETIVE OPERATORS

This section contains the interpretive operators for procedural statements. The procedural operators are listed in paragraph 5.1. This list contains the GOAL syntax diagram number, interpretive operator mnemonics and numbers, a concise statement of the operators functions and examples of GOAL statements for each operator.

5.1 PROCEDURAL INTERPRETIVE OPERATORS

The procedural operators define functions and provide data for the real time executive.

	Syntax Diagram Number	Operator Mnemonic	Code	Operator Function
5.1.1	01	ACTTAB	ØP01	Assign an active status to all rows of a table. ACTIVATE (ANY TABLE);
5.1.2	01	ACTROW	ØP02	Activate selected row(s) of specified table. ACTIVATE (TABLE) ROW 3;
5.1.3	02	APLPVO	ØP03	Apply the present value of one analog device to another. APPLY PRESENT VALUE OF <EXDES>1 TO <EXDES>2 ;
5.1.4	02	APLEXD	ØP04	Apply self defining analog stimuli to a system under test. APPLY <STAGE POWER>;
5.1.5	02	ADLDAT	ØP05	Apply an analog quantity to a system under test. APPLY 10V, 30V TO <PBUS1>, <PBUS 2>; APPLY 28V TO <EXDES 1>, <EXDES 2>;
5.1.6	03	ASSIGN	ØP06	Assign a state to an internal name. ASSIGN (SUPPLY) = ON;

	Syntax Diagram Number	Operator Mnemonic	Code	Operator Function
5.1.7	04	AVRAGE	ØP07	Average a measurement or group of measurements of a system under test. AVERAGE 5 READINGS OF <EXDES 1> AND SAVE AS (PRESS);
5.1.8	07	BGNPGM	ØP08	Initialize for program execution. BEGIN PROGRAM (IU COOL) REVISION 3;
5.1.9	08	BGNSUB	ØP09	Initialize for subroutine execution. BEGIN SUBROUTINE (GSCU ON);
5.1.10	15	CONCNT	ØP12	Concurrently record the present value of an external designator to a test system device. CONCURRENTLY DISPLAY PRESENT VALUE OF <COOLANT TEMP> TO <CRT 1>, <CRT 2>;
	15	CONCNT	ØP12	Concurrently monitor specified measurements and display out of tolerance measurements on test system devices. CONCURRENTLY VERIFY <BATT 1>, <BATT 2> ARE BETWEEN 28V AND 32V AND DISPLAY EXCEPTIONS TO <CRT 1>;
	15	CONCNT	ØP12	Concurrently execute a test program. CONCURRENTLY PERFORM PROGRAM (IU COOL) REVISION 3;
5.1.11	26	DELAY	ØP13	Delay the execution of the next sequential statement for a specified time. DELAY 10 SECS;
	26	DELAY	ØP13	Conditionally delay the execution of the next sequential statement. DELAY UNTIL <CDC> IS EQUAL TO -30.0 SECS;

	Syntax Diagram Number	Operator Mnemonic	Code	Operator Function
5.1.11 (Cont)	26	DELAY	ØP13	Delay the execution of the next sequential statement for a specified time or until a specified condition is met. DELAY 10 SECS OR UNTIL <CDC> IS EQUAL TO -30.0 SECS;
5.1.12	28	DISABL	ØP14	Disable the interrupts enabled by When Interrupt statements. DISABLE STEP 120, STEP 200; DISABLE ALL;
5.1.13	29	END P/S	ØP15	Return control to the calling program or to the executive depending upon whether 'END' was encountered in a subroutine or a program, respectively. END PROGRAM; (SUBROUTINE)
5.1.14	34	GO TO	ØP16	Unconditionally branch to the specified statement then continue execution. GO TO STEP 330;
5.1.15	37	INHTAB	ØP17	Assign an inactive status to all rows of a table. INHIBIT (ANY TABLE);
5.1.16	37	INHROW	ØP18	Assign an inactive status to selected row(s) of table. INHIBIT (TABLE) ROW 3;
5.1.17	40	ISUPVO	ØP19	Issue the present digital pattern of one device to another. ISSUE PRESENT VALUE OF <SW SEL CH 12> TO <STAGE SEP>;

	Syntax Diagram Number	Operator Mnemonic	Code	Operator Function
5.1.18	40	ISUEXD	ØP20	Issue self defining digital patterns to a system under test. ISSUE <S4B SW SEL CH 59>;
5.1.19	40	ISUDAT	ØP21	Issue specified digital patterns to a system under test. ISSUE B1011 TO <PANEL 1>, <PANEL 2>;
5.1.20	42	LETEQU	ØP22	Assign a numeric value to an internal name. LET (A) = (A) + 1 LET (B) = ((A) + (C) **2) - 1.37;
5.1.21	43	OUTXCP	ØP23	Output default or specified messages to the default or specified devices when an error condition occurs. DISPLAY EXCEPTIONS USING MESSAGES FROM (MSG TABLE) COLUMN 3 TO <CRT 1>;
5.1.22	55	PFMPGM	ØP24	Return control and provide information to the executive for executing a GOAL program. PERFORM PROGRAM (IU COOL) REVISION 3;
5.1.23	56	PFMSUB	ØP25	Return control and provide information to the executive for executing a GOAL program. PERFORM SUBROUTINE (GSCU ON);
5.1.24	60	READ	ØP26	Acquire data from the system under test and save in an internal name. READ <CDC> AND SAVE AS (CDTIME);
5.1.25	61	RECPVO	ØP27	Record the present value of an external designator to a test system device. RECORD <CDC> TO <MAG 1> , <CRT 1>;

	Syntax Diagram Number	Operator Mnemonic	Code	Operator Function
5.1.26	61	RECDAT	ØP28	Record text information on test system devices. DISPLAY TEXT (PREPS COMPLETE) TO <CRT 1> , <CRT 3> , <CRT 9>;
5.1.27	63	RELEAS	ØP29	Place concurrent operations in an inactive status. RELEASE ALL; RELEASE STEP 150, STEP 340;
5.1.28	64	REPEAT	ØP30	Execute a statement or group of statements a specified number of times. REPEAT STEP 20; REPEAT STEP 20 THROUGH STEP 28 FOR 3 TIMES:
5.1.29	66	REQEST	ØP31	Request and acquire text information from a keyboard and save it as an internal name. REQUEST TEXT (SELECT ONBOARD), TEXT (COOLING PUMP), TEXT (P1 OR P2), FROM <CRT 3> AND SAVE AS (PUMPON);
5.1.30	70	SETPVO	ØP32	Apply the present state of one discrete device to another. SET PRESENT VALUE OF <SWITCH 1> TO <SWITCH 2> , <SWITCH 3>;
5.1.31	70	SETEXD	ØP33	Set a discrete device to a self defined state. SET <PANEL SWITCHES> ;
5.1.32	70	SETDAT	ØP34	Set a discrete device to a specified state for a specified time interval, then set the device to the compliment state. SET <POWER SWITCH> TO ON FOR 3 SECS;

	Syntax Diagram Number	Operator Mnemonic	Code	Operator Function
5.1.33	73	STEPNO	ØP35	Provide statement label information to the executive. GO TO STEP 100; : : S100 GOAL STATEMENT;
5.1.34	75	STOP	ØP36	Stop the execution of the test procedure. Accommodate manual intervention for restricted or unrestricted restart. STOP; STOP AND INDICATE RESTART LABELS STEP 100, STEP 200;
5.1.35	78	TERMIN	ØP37	Terminate execution of the program, subroutine or test system. Return control to the calling program or the executive. TERMINATE; TERMINATE SYSTEM;
5.1.36	80	TIMPFX	ØP38	Synchronize execution of a statement with an external clock. WHEN <CDC> IS -30 SECS THEN
5.1.36	83	VERIFY	ØP39	Test the specified conditions then either execute the statement or branch to the next sequential statement. VERIFY <CDC> IS LESS THAN -30 SECS ELSE DISPLAY EXCEPTIONS TO <CRT 9> AND
5.1.38	83	IF	ØP40	Test the specified conditions then either execute the statement or branch to the next sequential statement. IF (VALVE) IS OPEN THEN

	Syntax Diagram Number	Operator Mnemonic	Operator Code	Operator Function
5.1.39	84	WHNINT	ØP41	Enable an interrupt and identify its conditions. WHEN INTERRUPT <FLAG1> OCCURS
5.1.40	84	AFTINT	ØP42	Specify action to be performed following interrupt service. AND RETURN ; GO TO S100 ;

6.0 INTERPRETIVE DATA FORMAT

This section describes the format of the interpretive data generated by the Translator. This data may be obtained on a seven or nine track tape according to the selected user options.

This data is logically grouped into three sections:

- Program Control Block
- Resident Data Area
- Procedural Operators

All of these sections are written in the output file (tape) as one or more records of the user specified block size.

6.1 PROGRAM CONTROL BLOCK (PCB)

This section of the interpretive data is always given as the first record of the output file. It is 100 words in length, and thus can always be contained in a single record. The PCB contains the information required to process the remainder of the interpretive data. Its contents and format are given in Figure VII.

6.2 RESIDENT DATA AREA

This section of the interpretive data file contains information which must be resident (or readily available) to support the execution of the procedural operators. This information is given as one or more records in the interpretive data file immediately following the PCB. This area may contain the following:

- External Reference Table
- Function Designator I/O Table
- Function Designator Names Table
- Internal Names Table
- Data Definition Control Blocks
- Statement Label Table
- Internal Statement Number Table
- Dynamic Data Area

The location and quantity of each of the above is given in the PCB. The contents and format of these areas is given in Figures VIII through XIV.

6.3 INTERPRETIVE OPERATORS

This section describes the format of the operators generated by the GOAL Interpretive Code Translator. This description is given in the form of operator/parameter blocks. All of the operators described in Section 5.0 are included. For each operator a mnemonic name is given. This name is used for reference purposes only. The names were selected to provide a concise notation for the operators, and also to suggest the function which they perform.

The structure of the interpretive code operator blocks reflects the following design considerations:

- Maximum retention of information obtained during GOAL program compilation.
- Compact representation of operator parameters.
- Minimum processing requirements placed on real time executive.
- Standard formats used to represent frequently used parameters.

6.3.1 Operator Format Conventions

The first word of each operator block is the type code. This is identified in the following descriptions as OP nn. Each operator has a unique type code designated by the number nn. This is the actual value of the type code generated by the translator. However, these type codes may be modified by means of relatively simple changes to the translator routine, TXINIT.

The second word of each operator block is always the block length. This is given in terms of the number of output WORDS required to contain the block. For most operator types the length will vary depending upon the options selected in the source GOAL statement.

The remainder of the operator block is comprised of Counts, flags and fields which are identified in the description according to the following conventions:

External Designators - (EXTDES) - Three words are always required to specify an 'EXTERNAL DESIGNATOR' in an operator block.

1st WORD - No. of FD's

2nd WORD - A(INH) or 0

3rd WORD - A(FDLIST)

The first word always indicates the number of function designators contained in or referenced by the 'EXTERNAL DESIGNATOR'. Its value may range from 1 to 45.

If the second word is non-zero, it specifies the address in the resident data area of the inhibits array associated with the '(TABLE) FUNCTIONS'. In this case the third word is the address, in the resident area, where a list of function designator pointers is located.

If the second word is zero, the third word specifies the address, relative to the beginning of the operator block, where the list of function designator pointers is located.

In both cases, the pointers specify the address in the resident function designator I/O table where the type, hardware address, and symbolic name may be determined.

Internal Names - (INTNAM) - All data references, whether variables or constants, require the use of the INTNAM field in the operator blocks. This field may vary in length from two to four words. Its format is as follows:

1st WORD - Type

2nd WORD - A(DDCB)

3rd WORD - A(SUBSCRIPT) - (Type 2 or 3)

4th WORD - A(SUBSCRIPT) - optional (Type 3)

The first word always contains 1, 2 or 3. A Type 1 INTNAM has a length of 2 and uses no subscripts. It corresponds to a single data item, list, or table. A Type 2 INTNAM has a length of 3 and corresponds to a list entry or a table column. A Type 3 is always a Table entry.

The second word contains the location of the Data Definition Control Block, (DDCB), which is associated with the data. If this word contains a positive value, the DDCB is located in the resident table area. If this word is negative its value indicates the position of the DDCB in the current operator block.

Words three and four specify the location of words which contain integer values of the subscripts. If the location value is positive, the subscript word is located in the resident data area. If the value is negative the subscript is located in the current operator block.

The DDCB contents and format are given in Figure VIII.

Addressing - Unless otherwise noted all addresses with a positive value refer to locations in the resident area. Those with a negative value refer to locations in the current operator block. A (INTNAM) always refers to the current operator block.

Optional Area - The portion of the operator blocks indicated as OPTIONAL may contain:

- o INTNAM's
- o EXTDES
- o DDCB's
- o Fixed Data

6.3.2 Operator Block Formats

```
ACTTAB   WORD 1 - OPCODE = OP01 (ACTIVATE) - TABLE
         WORD 2 - LENGTH = 0004
         WORD 3 - A(INHIBITS)
         WORD 4 - NUMBER OF INHIBITS

ACTROW   WORD 1 - OPCODE = OP02 (ACTIVATE) - ROWS
         WORD 2 - LENGTH = VARIABLE
         WORD 3 - A(INHIBITS)
         WORD 4 - NUMBER OF ROWS
         WORD 5 - A(ROW NO.) OR -(ROW NO)
         THRU N -

APLPVO   WORD 1 - OPCODE = OP03 (APPLY PVO)
         WORD 2 - LENGTH = VARIABLE
         WORD 3 - NO. OF FD'S (1ST-SENSOR) - EXTDES
         WORD 4 - A(INH) OR 0 (1ST-SENSOR)
         WORD 5 - A(FD LIST)  (1ST-SENSOR)
         WORD 6 - NO. OF FD's (2ND-LOAD) - EXTDES
         WORD 7 - A(INH) OR 0 (2ND-LOAD)
         WORD 8 - A(FD LIST)  (2ND-LOAD)
         THRU N - OPTIONAL
```

```
APLEXD    WORD 1 - OPCODE = OP04 (APPLY EXD)
          WORD 2 - LENGTH - VARIABLE
          WORD 3 - NO. OF FD'S - EXTDES
          WORD 4 - A(INH) OR 0
          WORD 5 - A(FD LIST)
          THUR N - OPTIONAL

APLDAT    WORD 1 - OPCODE = OP05 (APPLY DATA)
          WORD 2 - LENGTH = VARIABLE
          WORD 3 - NO. OF FD'S (LOAD) - EXTDES
          WORD 4 - A(INH) OR 0 (LOAD)
          WORD 5 - A(FD LIST)  (LOAD)
          WORD 6 - NO. OF INTNAM'S
          THRU ? - A(INTNAM)
          THRU N - OPTIONAL

ASSIGN    WORD 1 - OPCODE = OP06 (ASSIGN
          WORD 2 - LENGTH = VARIABLE
          WORD 3 - NAM TYPE (1ST) - INTNAM
          WORD 4 - A(DDCB)  (1ST)
          WORD 5 - OPT OR 0 (1ST)
          WORD 6 - OPT OR 0 (1ST)
          WORD 7 - NAM TYPE (2ND) - INTNAM
          WORD 8 - A(DDCB)  (2ND)
          THRU N - OPT      (2ND)

AVRAGE    WORD 1 - OPCODE = OP07 (AVERAGE)
          WORD 2 - LENGTH = VARIABLE
          WORD 3 - NO. OF SAMPLES
          WORD 4 - NO. OF FD'S (SENSOR) - EXTDES
          WORD 5 - A(INH) OR 0 (SENSOR)
          WORD 6 - A(FD LIST)  (SENSOR)
          WORD 7 - NAM TYPE - INTNAM
          WORD 8 - A(DDCB)
          THRU N - OPTIONAL

BGNPGM    WORD 1 - OPCODE = OP08 (BEGIN PROGRAM)
          WORD 2 - LENGTH = 0002

BGNSUB    WORD 1 - OPCODE = OP09 (BEGIN SUBROUTINE)
          WORD 2 - LENGTH = VARIABLE
          WORD 3 - NO. OF PARMS OR 0
          THRU ? - NAM/FD = 0/1          * ALWAYS
          THRU ? - A(DDCB) OR A(FDIOT)   * PAIRS
          THRU N - OPTIONAL
```

```
CONCNT    WORD 1 - OPCODE = OP12 (CONC)
          WORD 2 - LENGTH = VARIABLE
          WORD 3 - A(INTNAM) OR 0 - TIME VALUE
          THRU N - OPTIONAL

DELAY     WORD 1 - OPCODE = OP13 (DELAY).
          WORD 2 - LENGTH = VARIABLE
          WORD 3 - A(INTNAM) OR 0 - TIME VALUE
          WORD 4 - NO. OF FD'S OR 0 - EXTDES
          WORD 5 - A(INH) OR 0 (OPT)
          WORD 6 - A(FD LIST)  (OPT)
          WORD 7 - RELATIONSHIP (OPT)
          WORD 8 - A(INTNAM)    (OPT)
          WORD 9 - A(INTNAM)    (OPT) IF REQ'D
          THRU N - OPTIONAL

DISABL    WORD 1 - OPCODE = OP14 (DISABLE INTERRUPT)
          WORD 2 - LENGTH = VARIABLE
          WORD 3 - NO. OF LABELS - (0 IMPLIES ALL)
          THRU N - A(LABEL ADR'S)

ENDP/S    WORD 1 - OPCODE = OP15 (END PROGRAM/SUBROUTINE)
          WORD 2 - LENGTH = 0002

GOTO      WORD 1 - OPCODE = OP16 (GO TO)
          WORD 2 - LENGTH = 0003
          WORD 3 - A(LBL ENT)

INHTAB    WORD 1 - OPCODE = OP17 (INHIBIT) - TABLE
          WORD 2 - LENGTH = 0004
          WORD 3 - A(INHIBITS)
          WORD 4 - NUMBER OF INHIBITS

INHROW    WORD 1 - OPCODE = OP18 (INHIBIT) - ROWS
          WORD 2 - LENGTH = VARIABLE
          WORD 3 - A(INHIBITS)
          WORD 4 - NUMBER OF ROWS
          WORD 5 - A(ROW NO.) OR -(ROW NO.)
          THRU N - SAME

ISUPVO    WORD 1 - OPCODE = OP19 (ISSUE PVO)
          WORD 2 - LENGTH = VARIABLE
          WORD 3 - NO. OF FD'S (1ST-SENSOR) - EXTDES
          WORD 4 - A(INH) OR 0 (1ST-SENSOR)
          WORD 5 - A(FD LIST)  (1ST-SENSOR)
          WORD 6 - NO. OF FD'S (2ND-LOAD) - EXTDES
          WORD 7 - A(INH) OR 0 (2ND-LOAD)
          WORD 8 - A(FD LIST)  (2ND-LOAD)
          THUR N - OPTIONAL
```

```
ISUEXD  WORD 1 - OPCODE = OP20 (ISSUE EXD)
        WORD 2 - LENGTH - VARIABLE
        WORD 3 - NO. OF FD'S - EXTDES
        WORD 4 - A(INH) OR 0
        WORD 5 - A(FD LIST)
        THRU N - OPTIONAL

ISUDAT  WORD 1 - OPCODE = OP21 (ISSUE)
        WORD 2 - LENGTH = VARIABLE
        WORD 3 - NO. OF FD'S (LOAD) - EXTDES
        WORD 4 - A(INH) OR 0 (LOAD)
        WORD 5 - A(FDLIST)   (LOAD)
        WORD 6 - NO. OF INTNAM'S
        THRU ? - A(INTNAM)
        THRU N - OPTIONAL

LETEQU  WORD 1 - OPCODE = OP22 (LET EQUAL)
        WORD 2 - LENGTH = VARIABLE
        WORD 3 - A(INTNAM) - OBJECT
        WORD 4 - NO. OF ENTRIES IN LIST
        THRU ? - ENTRY - A(INTNAM) OR OPERATOR
        THRU N - OPTIONAL

OUTXCP  WORD 1 - OPCODE = OP23 (OUTPUT EXCEPTION)
        WORD 2 - LENGTH = VARIABLE
        WORD 3 - NO. OF FD'S (SYSTEM) - EXTDES
        WORD 4 - A(INH) OR 0 (SYSTEM)
        WORD 5 - A(FDLIST)   (SYSTEM)
        WORD 6 - INTNAM TYPE - (0 FOR DEFAULT) - INTNAM
        WORD 7 - A(DDCB)     - (0 FOR DEFAULT)
        THRU N - OPTIONAL

PFMPGM  WORD 1 - OPCODE = OP24 (PERFORM PROGRAM)
        WORD 2 - LENGTH = 0003
        WORD 3 - A(EXT REF)

PFMSUB  WORD 1 - OPCODE = OP25 (PERFORM SUBROUTINE)
        WORD 2 - LENGTH = VARIABLE
        WORD 3 - A(EXT REF)
        WORD 4 - CRITICAL FLAG NORM/CRIT = 0/1
        WORD 5 - NO. OF PARMS OR 0
        THRU ?   A(PARM)
        THRU ?   PARM = INTNAM OR EXDES
        THRU N   OPTIONAL
```

```
READ      WORD 1 - OPCODE = OP26 (READ)
          WORD 2 - LENGTH = VARIABLE
          WORD 3 - NO. OF FD'S (SENSOR) - EXTDES
          WORD 4 - A(INH) OR 0 (SENSOR)
          WORD 5 - A(FD LIST)  (SENSOR)
          WORD 6 - NAM TYPE - INTNAM
          WORD 7 - A(DDCB)
          THRU N - OPT

RECPVO    WORD 1 - OPCODE = OP27 (RECORD PVO)
          WORD 2 - LENGTH = VARIABLE
          WORD 3 - NO. OF FD'S (SENSOR) - EXTDES
          WORD 4 - A(INH) OR 0 (SENSOR)
          WORD 5 - A(FD LIST)  (SENSOR)
          WORD 6 - NO. OF FD'S (SYSTEM) - EXTDES
          WORD 7   A(INH) OR 0 (SYSTEM)
          WORD 8   A(FD LIST)  (SYSTEM)
          THRU N   OPTIONAL

RECDAT    WORD 1 - OPCODE = OP28 (RECORD DATA)
          WORD 2 - LENGTH - VARIABLE
          WORD 3 - NO. OF FD'S - EXTDES
          WORD 4 - A(INH) OR 0
          WORD 5 - A(FD LIST)
          WORD 6 - NO. OF INTNAM'S
          THRU ? - A(INTNAM)
          THRU N - OPTIONAL

RELEAS    WORD 1 - OPCODE = OP29 (RELEASE CONCURRENT)
          WORD 2 - LENGTH = VARIABLE
          WORD 3 - NO. OF STMT'S - (0 FOR ALL)
          THRU N - A(LBL ADR)

REPEAT    WORD 1 - OPCODE = OP30 (REPEAT)
          WORD 2 - LENGTH = 0005
          WORD 3 - NO. TIMES
          WORD 4 - A(LBL) - 1ST
          WORD 5 - A(LBL) - 2ND (OR 0)

REQEST    WORD 1 - OPCODE = OP31 (REQUEST)
          WORD 2 - LENGTH = VARIABLE
          WORD 3 - A(FD)
          WORD 4 - A(INTNAM) - INPUT DATA
          WORD 5 - NO. OF INTNAM'S - MESSAGES
          THRU ? - A(INTNAM) - MESSAGES
          THRU N - OPTIONAL
```

```
SETPVO    WORD 1 - OPCODE = OP32 (SET PVO)
          WORD 2 - LENGTH = VARIABLE
          WORD 3 - NO. OF FD'S (1ST-SENSOR) - EXTDES
          WORD 4 - A(INH) OR 0 (1ST-SENSOR)
          WORD 5 - A(FD LIST)  (1ST-SENSOR)
          WORD 6 - NO. OF FD'S (2ND-LOAD) - EXTDES
          WORD 7 - A(INH) OR 0 (2ND-LOAD)
          WORD 8 - A(FD LIST)  (2ND-LOAD)
          THRU N - OPTIONAL

SETEXD    WORD 1 - OPCODE = OP33 (SET EXD)
          WORD 2 - LENGTH = VARIABLE
          WORD 3 - A(INTNAM) OR 0 - TIME VALUE
          WORD 4 - NO. OF FD'S - EXTDES
          WORD 5 - A(INH) OR 0
          WORD 6 - A(FD LIST)
          THRU N - OPTIONAL

SETDAT    WORD 1 - OPCODE = OP34 (SET DATA)
          WORD 2 - LENGTH = VARIABLE
          WORD 3 - A(INTNAM) OR 0 - TIME VALUE
          WORD 4 - NO. OF FD'S - EXTDES
          WORD 5 - A(INH) OR 0
          WORD 6 - A(FD LIST)
          WORD 7 - NO. OF INTNAM'S - DATA
          THRU ? - A(INTNAM)
          THRU N - OPTIONAL

STEPNO    WORD 1 - OPCODE = OP35 (STATEMENT LABEL)
          WORD 2 - LENGTH = 0003
          WORD 3 - RESERVED

STOP      WORD 1 - OPCODE = OP36 (STOP)
          WORD 2 - LENGTH - VARIABLE
          WORD 3 - NO. OF LBL'S - (0 FOR NONE)
          THRU N - A(LBL ADR)

TERMIN    WORD 1 - OPCODE = OP37 (TERMINATE)
          WORD 2 - LENGTH = 0003
          WORD 3 - FLAG - NORM/SYS = 0/1

TIMPFX    WORD 1 - OPCODE = OP38 (TIME PREFIX)
          WORD 2 - LENGTH = VARIABLE
          WORD 3 - AFTER/WHEN = 0/1
          WORD 4 - A(FD)
          WORD 5 - INTNAM TYPE - INTNAM
          WORD 6 - A(DDCB)
          THRU N - OPTIONAL
```

```
VERIFY    WORD 1 - OPCODE = OP39 (VERIFY)
          WORD 2 - LENGTH = VARIABLE
          WORD 3 - ELSE/THEN = 1/3
          WORD 4 - A(INTNAM) OR 0 (TIME LIMIT)
          WORD 5 - NO. OF FD'S - EXTDES
          WORD 6 - A(INH) OR 0
          WORD 7 - A(FD LIST)
          WORD 8 - RELATIONSHIP
          WORD 9 - A(INTNAM) - COMPARAND
          WORD 10- A(INTNAM) - COMPARAND - IF REQ'D
          THRU N - OPTIONAL

IF        WORD 1 - OPCODE = OP40 (IF)
          WORD 2 - LENGTH = VARIABLE
          WORD 3 - RELATIONSHIP
          WORD 4 - A(INTNAM) - OBJECT
          WORD 5 - A(INTNAM) - 1ST COMPARAND
          THRU 6 - OPTIONAL - 2ND COMPARAND
          THRU N - OPTIONAL

WHNINT    WORD 1 - OPCODE = OP41 (WHEN INTERRUPT)
          WORD 2 - LENGTH = 0003
          WORD 3 - A(FD)

AFTINT    WORD 1 - OPCODE = OP42 (AFTER INTERRUPT)
          WORD 2 - LENGTH = 0003
          WORD 3 - A(LBL ADR) OR 0
```

7.0 PROCESSING TECHNIQUES

This section describes the processing of inputs and outputs, and the general program structure of the GOAL Interpretive Code Translator. The principal modules of the program are identified and described. Special considerations regarding programming languages and hardware characteristics are discussed.

7.1 PROGRAM STRUCTURE AND PROCESSING

The general processing flow for the Translator program is shown in Figures I and II, and described in Section 2.0 of this document. The principal modules of the program are identified in Figure XVI. To minimize core requirements for execution of the Translator, most modules of the program are designed for execution as overlays. Following is a brief description of the principal modules:

TXMAIN - This module is the Translator mainline program. Its principal function is to sequence the execution of the more specialized Translator modules. This module contains the initial entry point which is given control by the system at execution time. Direct access files which are used by other Translator routines are defined in this module.

TXINIT - This module is the Translator initialization routine. It is called by TXMAIN as the first step in translation processing. The functions of TXINIT include:

- Establish the size of tables used by the Translator
- Load symbol table data file
- Read and process option control cards
- Initialize flags, counters, etc.

TXINIT will validate all user option control cards. If any errors are detected a diagnostic message is given and the run is terminated. In all cases a listing of both selected and default user options is generated.

TXPREP - This module is executed immediately after TXINIT. Its principal function is to build and initialize the tables and control blocks used by operator generation routines. These tables include:

- Program Control Block
- External Reference Table
- Function Designator I/O Table
- Function Designator Names Table

- Internal Names Table
- Data Definition Control Blocks
- Statement Label Table
- Internal Statement Number Table
- Dynamic Data Area

TXINPT - This module is the Translator input routine which is used to read each record of the 'Intermediate GOAL' data file. This file is generated by the GOAL Compiler and processed by the Translator to generate the interpretive data. Each record is read into a common input buffer area where it is available to other Translator modules for processing. The record type code is placed in a special location where it is available to TXMAIN for use in selecting the appropriate routines for further processing.

TXnn - This is a series of modules titled TX01, TX02, ..., TXnn. These routines are called by TXMAIN to perform processing of specific 'Intermediate GOAL' data records. The two digit number, nn, corresponds to the type code of the record which is processed. The functions of these routines include:

- Symbol table updates - information is stored in the symbol table to enable address resolution for branching and data reference.
- Storage allocation - space is identified and allocated for the internal data used by the GOAL program.
- Operator/parameter generation - a tabular representation of the interpretive operators and parameters is generated.

Data record formats are described in the document, GOAL Compiler Intermediate Text Output Formats.

TXLIST - This module controls the generation of the optional output listings produced by the Translator. Its principal function is to scan the output request flags and initiate the execution of the routines which generate specific reports.

TXLnn - This is a series of modules titled TXL01, TXL02, ..., TXLnn. These routines are called by TXLIST to generate specific output reports. The two digit number, nn, corresponds to the listing described in Section 4.0, paragraph number nn.

TX4MAT - This module is used to generate the Interpretive Data output of the Translator. This file is constructed according to the options which are established by TXINIT. This routine will call special purpose assembler language modules to perform detailed bit manipulation.

7.2 SPECIAL CONSIDERATIONS

Since the Translator program manipulates data at the 'bit' level, several aspects of its operation will be affected by the programming language and hardware characteristics of the CPU in which it is executed.

Programming Language - As directed by NASA, maximum use will be made of FORTRAN in programming the Translator. This does not pose prohibitive difficulties except for cases where data must be manipulated at the 'bit' level. An example of this is when a binary field is to be moved to a non-word-boundary position. Since FORTRAN provides no practical method to accomplish this, special purpose assembler language routines will be required.

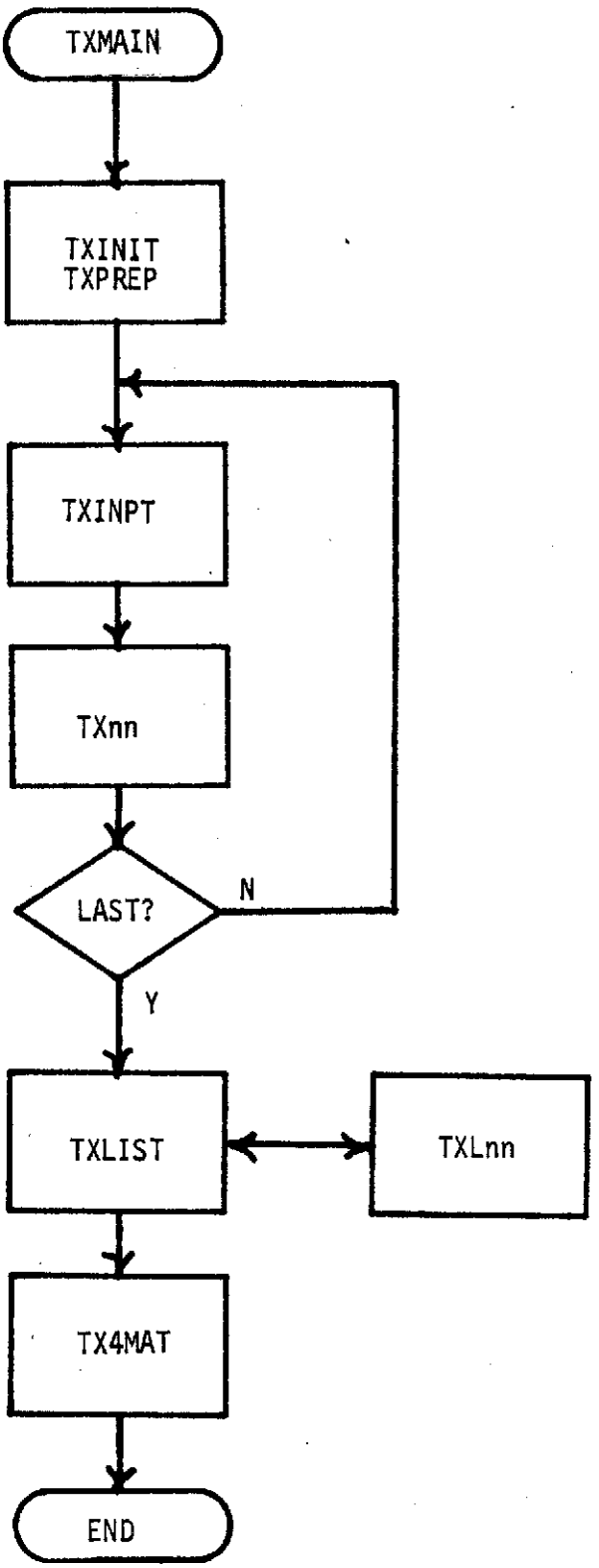
CPU Characteristics - A necessary step in the translation process is the generation of a tabular version of the interpretive data stream. This tabular data file is logically equivalent to the Translator output file, but it conforms to the characteristics of the CPU in which the translator is executed. This tabular file is reformatted to produce the Translator interpretive data output. Thus, some of the interpretive code output options will be constrained by the CPU characteristics of the machine on which the Translator program is executed. The principal constraints include:

- Target word size - 32 bits max.
- Target char/word - 4 max.

The following Translator routines were written in System/360 Basic Assembler Language:

1. MOVCHR
2. TXNCON
3. TXCCON
4. TXTAPE

The remainder of this section contains descriptions of individual GOAL Translator routines.



INTERPRETIVE TRANSLATOR PROGRAM STRUCTURE

FIGURE XVI

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXMAIN

FUNCTION – To control the sequence of execution of the translator subroutines.

CALLED BY –

SUBROUTINES CALLED – TXINIT, TXPREP, TXINPT, TXLIST, TX4MAT, EXIT and the TXnn routines which develop the interpretive operator codes.

DESCRIPTION – This is the mainline, highest level translator program. It controls the sequence of execution of the lower level routines.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXINIT

FUNCTION – To read and process the translator control cards and to initialize all flags and tables.

CALLED BY – TXMAIN

SUBROUTINES CALLED – EXIT

DESCRIPTION – Read and process the master card, list control card, character set card(s), invalid operator card(s). Set all flags according to input cards. Set up the ASCII conversion table. Set up the operator table. Read in the symbol table.

During the processing of control cards, if any error occurs, the job will be terminated.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXPREP

FUNCTION – To prepare the Program Control Block and 'resident area' tables for the operator generation phase.

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXPUT, MOVCHR, TXGET

DESCRIPTION – Initial values are assigned to Program Control Block addresses and counters. The Symbol Table is processed to generate the selected resident area tables.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXINPT

FUNCTION – To read a record from the Intermediate GOAL, (TXT), data file.

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXPUT

DESCRIPTION – The Intermediate GOAL (TXT) record is read and placed in 'COMMON'. The Internal Statement Number table is updated, if required.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXLIST

FUNCTION – To control the sequence of execution of the individual list programs.

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXL01, TXL02, TXL03, TXL04, TXL05, TXL07, TXL08, TXL09, TXL11, TXL12.

DESCRIPTION – This program tests the list option control flags and initiates the execution of the lower level list program in a predetermined sequence.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX4MAT

FUNCTION – To generate the Translator Output Tape.

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXTAPE, TXGET, TXCCON, TXPUT

DESCRIPTION – The operation is performed in three steps.

1. The Program Control Block record is written.
2. Symbolic Text tables are converted and the 'resident area' is written.
3. Procedural operator records are written.

Data required to support these steps is obtained from 'COMMON' and the tabular file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXL01

FUNCTION – To list the contents of the Program Control Block of the Tabular Program file.

CALLED BY – TXLIST

SUBROUTINES CALLED – NONE

DESCRIPTION – This program lists the Program Control Block by TABULAR FILE address and contents. The user list option flags, the specification for the output program tape, the file address and number of entries in the various tables, data areas and operator code block is printed.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXL02

FUNCTION – To list the contents of the Internal Names Table of the tabular program file.

CALLED BY – TXLIST

SUBROUTINES CALLED – TXGET

DESCRIPTION – This program lists the tabular file addresses and contents of the Internal Names Table. Three items are printed for each entry in the table.

1. The number of characters in the name
2. The internal name
3. The address of the data definition control block for the name.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXL03

FUNCTION – To list the contents of the Statement Label Table of the tabular program file.

CALLED BY – TXLIST

SUBROUTINES CALLED – TXGET

DESCRIPTION – This program lists the tabular file addresses and content of the table. Two items are printed for each entry in the table.

1. The step number
2. The address of the step number in the Operator Code Block.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXL04

FUNCTION – To list the contents of the Internal Statement Number Table of the tabular program file.

CALLED BY – TXLIST

SUBROUTINES CALLED – TXGET

DESCRIPTION – This program lists the tabular file addresses and contents of the table. One item is printed for each entry in the table.

1. Address of the Internal Statement Number.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXL05

FUNCTION – To list the contents of the Function Designator Names Table of the tabular program file.

CALLED BY – TXLIST

SUBROUTINES CALLED – TXGET

DESCRIPTION – This program lists the tabular file addresses and contents of the table. Three items are printed for each entry in the table.

1. The number of characters in the name
2. The name
3. The address of the data definition control block for the name.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXL07

FUNCTION – To list the contents of the Data Area of the tabular program file.

CALLED BY – TXLIST

SUBROUTINES CALLED – TXGET

DESCRIPTION – This program prints the tabular file addresses and contents of the Data Area. This is essentially a core dump of the data.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXL08

FUNCTION – To list the contents of the External Reference Table of the tabular program file.

CALLED BY – TXLIST

SUBROUTINES CALLED – TXGET

DESCRIPTION – This program prints the tabular file addresses and contents of the table. Four items are printed for each entry.

1. Identification Code Number
2. Address of the Number.
3. The number of characters in the External Reference name.
4. The External Reference name.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXL09

FUNCTION – To list the contents of the Tabular Program File.

CALLED BY – TXLIST

SUBROUTINES CALLED – TXGET

DESCRIPTION – This program prints the tabular file addresses and content of the tabular program file. This is essentially a core dump of the file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXL11

FUNCTION – To list the contents of the Function Designator I/O Table of the tabular program file.

CALLED BY – TXLIST

SUBROUTINES CALLED – TXGET

DESCRIPTION – This program prints the tabular file addresses and contents of the table. Two items are printed for each entry.

1. The I/O Type Code
2. Address of the Code.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXL12

FUNCTION – To list the contents of the Data Definition Control Block of the tabular program file.

CALLED BY – TXLIST

SUBROUTINES CALLED – TXGET

DESCRIPTION – This program prints the tabular file addresses and contents of the block. There are twelve formats used, one for each of the 12 types of declarative data.

1. Numeric data	7. State data
2. Numeric list	8. State list
3. Numeric table	9. State table
4. Quantity data	10. Text data
5. Quantity list	11. Text list
6. Quantity table	12. Text table

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXGET

FUNCTION – To retrieve a block of data from the tabular file.

CALLED BY – Any Translator Routine

SUBROUTINES CALLED – None

DESCRIPTION – Three parameters are required:

- o Data Area
- o Data Length
- o Tabular File Address

The specified data is read from the tabular file and moved to the indicated Data Area.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXPUT

FUNCTION – To enter a block of data into the tabular file.

CALLED BY – Any Translator Routine

SUBROUTINES CALLED – None

DESCRIPTION – Three parameters are required:

- o Data Area
- o Data Length
- o Tabular File Address

The specified data is written in the tabular file at the indicated address. The address word is incremented by the Data Length.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – MOVCHR

FUNCTION – To block characters according to the selected value of 'Characters per Word', (CPW).

CALLED BY – TX08, TX16, TX63, TXPREP

SUBROUTINES CALLED – None

DESCRIPTION – This routine is written in 360 Assembler Language. Linkage with the calling FORTRAN programs is accomplished via standard calling sequence. Three parameters are used:

- o Address of a character
- o Address of a word
- o Target character position in word

The specified character is moved to the indicated position in word.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXNCON

FUNCTION – To convert a System 360 floating point word to 'integer' + 'exponent' format.

CALLED BY – TX02, TX04, TX10, TX12, TX19, TX21

SUBROUTINES CALLED – None

DESCRIPTION – This routine is written in System/360 Assembler Language. Four parameters are required:

- o Floating Point Value
- o Word for 'integer' part
- o Word for 'exponent' part
- o Bits per output word

The floating point word is 'disected' and the 'integer' and 'exponent' values are set. The integer part is shifted to achieve a zero exponent, if possible.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXPUTL

FUNCTION – To enter a DDCB/Data or function designator list in the literal reference file

CALLED BY – TX02, TX04, TX06, TX08, TX18

SUBROUTINES CALLED – None

DESCRIPTION – Three parameters are required:

- o Symbol reference number
- o Data Area
- o Data Length

The file is updated and a directory entry is generated for the specified entry.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXGETL

FUNCTION – To retrieve a DDCB/DATA or function designator list from the literal reference file.

CALLED BY – TXXGEN, TXNGEN

SUBROUTINES CALLED – None

DESCRIPTION – Three parameters are required:

- o Symbol reference number
- o Data Area
- o Data Length

The specified symbol is located in the directory and the data is read from the literal reference file. The data and length are moved to the specified areas.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXLOOK

FUNCTION – To identify symbolic references in the GOAL compiler symbol table.

CALLED BY – Any Translator Routine

SUBROUTINES CALLED – None

DESCRIPTION – Two parameters are required:

- o Internal symbol reference number
- o Condition code

The symbol table is searched for the specified symbolic reference. If found, the Condition code is set to its address. Otherwise, the Condition code is set to -1.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXOPUT

FUNCTION – To output an operator block format to the tabular file and/or output listings.

CALLED BY – Most 'TXnn' Routines

SUBROUTINES CALLED – TXPUT, TXERR, FORTRAN I/O

DESCRIPTION – The operator block is located in a 'COMMON' area. The buffer contents are checked and if there is not sufficient room it is padded with zero's and a new block is started. The opcode is tested. If invalid, a warning is given and tape generation suppressed. If the Operator List option is selected the operator is is listed. In any case the operator is written in the tabular file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXERR

FUNCTION – To process terminal errors.

CALLED BY – Any Translator Routine

SUBROUTINES CALLED – System 'EXIT', FORTRAN I/O

DESCRIPTION – One parameter is required:

- o Error code number

A diagnostic message indicating the error code number is printed and the run is terminated. This routine is used to process 'unrecoverable' errors and conditions.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXNAML

FUNCTION – To determine the size of the INTNAM format associated with a reference in the Intermediate TXT.

CALLED BY – Most 'TXnn' Routines

SUBROUTINES CALLED – TXERR

DESCRIPTION – Two parameters are required:

- o Intermediate TXT 'intnam' field
- o Word to contain the length

The specified field is examined and the required length is stored in the indicated word.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXNGEN

FUNCTION – To generate the 'INTNAM' format in the current operator block and establish linkage to other required control blocks.

CALLED BY – Most of the 'TXnn' routines

SUBROUTINES CALLED – TXLOOK, TXGET, TXGETL, TXERR

DESCRIPTION – Two parameters are required:

- o Intermediate TXT 'intnam' field
- o Position in current operator block

The specified 'intnam' is identified according to type and usage. The INTNAM format is generated. If the usage is a literal constant, a DDCB and the associated data are generated in the current operator block.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXXGEN

FUNCTION – To generate the EXTDES format in the current operator block, and to establish linkage to other required control blocks.

CALLED BY – Most 'TXnn' Routines

SUBROUTINES CALLED – TXLOOK, TXGET, TXGETL, TXERR

DESCRIPTION – Two parameters are required:

- Intermediate TXT 'extdes' field
- Position in current operator block

The specified 'extdes' is identified according to type and usage. The EXTDES format is generated. If the usage is not 'table functions' a list of function designator I/O table addresses is included in the current operator block.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXTAPE

FUNCTION – To write seven or nine track records according to specified user options.

CALLED BY – TX4MAT

SUBROUTINES CALLED – System/360 I/O Macros

DESCRIPTION – This routine is written in System/360 Assembler Language. Linkage with the calling FORTRAN program is accomplished via standard calling sequence. Five parameters are required:

- o Tabular Data Block
- o Tabular Word Count
- o Words per Tape Record
- o Bits per Tape Word
- o Seven/Nine Track Selection

The Tabular Data is converted to appropriate format and written on the specified tape.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TXCCON

FUNCTION – To convert EDCDIC characters to the specified character set.

CALLED BY – TX4MAT, TX08, TX16, TX63

SUBROUTINES CALLED – None

DESCRIPTION – This routine is written in System/360 Assembler Language. Five parameters are required:

- o Tabular Word (TEXT)
- o Characters per word
- o ASCII buffer (character set)
- o Bits per character
- o Bits per word

All characters in the tabular word are converted and positioned according to user options.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX02

FUNCTION – To process the GOAL type 02 intermediate text record - Declare Numeric Data (Initialized)

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXNCON, TXLOOK, TXGET, TXPUT, TXPUTL

DESCRIPTION – The numeric value specified in the type 2 record is converted to appropriate format. If the symbol is an internal name this value is stored in its associated dynamic data area. Otherwise, the symbol is placed in the literal file to enable its use as fixed data.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX04

FUNCTION – To process the GOAL type 04 intermediate text record - Declare Quantity Data (Initialized)

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXNCON, TXLOOK, TXGET, TXPUT, TXPUTL

DESCRIPTION – The numeric value specified in the type 4 record is converted to appropriate format. If the symbol is an internal name this value is stored in its associated dynamic data area and the DDCB units code is set. Otherwise the symbol is placed in the literal file to enable its use as fixed data.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX06

FUNCTION – To process the GOAL type 06 intermediate text record - Declare State (Initialized)

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXLOOK, TXGET, TXPUT, TXPUTL

DESCRIPTION – The logical state specified in the type 6 record is converted to appropriate format. If the symbol is an internal name it is stored in its associated dynamic data area. Otherwise, the symbol is placed in the literal file to enable its use as fixed data.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX08

FUNCTION – To process the GOAL type 08 intermediate text record - Declare Text (Initialized)

CALLED BY – TXMAIN

SUBROUTINES CALLED – MOVCHR, TXLOOK, TXGET, TXPUT, TXPUTL

DESCRIPTION – The text constant specified in the type 8 record is converted to appropriate format. If the symbol is an internal name the characters are moved to its associated dynamic data area. Otherwise, the symbol is placed in the literal file to enable its use as fixed data.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX10

FUNCTION – To process the GOAL type 10 intermediate text record - Declare Numeric List (Initialized)

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXNCON, TXLOOK, TXGET, TXPUT

DESCRIPTION – The numeric list data specified in the type 10 record is converted to appropriate format. The resulting data is moved to the dynamic data area associated with the indicated internal name.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX12

FUNCTION – To process the GOAL type 12 intermediate text record - Declare Quantity List (Initialized)

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXNCON, TXLOOK, TXGET, TXPUT

DESCRIPTION – The numeric data specified in the type 12 record is converted to appropriate format. This data is then moved to the dynamic data area associated with the indicated internal name. The units code(s) are updated in the DDCB.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX14

FUNCTION – To process the GOAL type 14 intermediate text record - Declare State List (Initialized)

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXLOOK, TXGET, TXPUT

DESCRIPTION – The logical state list is converted to appropriate format and moved to the dynamic data area associated with the indicated internal name.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX16

FUNCTION – To process the GOAL type 16 intermediate text record - Declare Text List (Row Initialization)

CALLED BY – TXMAIN

SUBROUTINES CALLED – MOVCHR, TXLOOK, TXGET, TXPUT

DESCRIPTION – The text list characters specified in the type 16 record are converted to appropriate format and moved to the dynamic data area associated with the indicated internal name and entry number.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX18

FUNCTION – To process the GOAL type 18 intermediate text record - Function Designator Array

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXERR, TXGET, TXPUTL

DESCRIPTION – The function designator string specified in the type 18 record is placed in the literal file to enable its use as fixed data.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX19

FUNCTION – To process the GOAL type 19 intermediate text record - Declare Numeric Table (Row Initialization)

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXNCON, TXLOOK, TXGET, TXPUT

DESCRIPTION – The numeric data specified in the type 19 record is converted to appropriate format and stored in the dynamic data area associated with the indicated internal name and row number.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX21

FUNCTION – To process the GOAL type 21 intermediate text record - Declare Quantity Table (Row Initialized)

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXNCON, TXLOOK, TXGET, TXPUT

DESCRIPTION – The numeric data specified in the type 21 record is converted to appropriate format and stored in the dynamic data area associated with the indicated internal name and row number. The units code(s) of the DDCB are updated.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX23

FUNCTION – To process the GOAL type 23 intermediate text record - Declare State Table (Row Initialization)

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXLOOK, TXGET, TXPUT

DESCRIPTION – The logical state(s) specified in the type 23 record are converted to appropriate format and stored in the dynamic data area associated with the indicated internal name and row number.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX24

FUNCTION – To process the GOAL type 24 intermediate text record - Repeat Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXERR, TXGET, TXOPUT

DESCRIPTION – The repetition count is moved to the operator buffer. The indicated labels are looked up in the label table and their respective addresses are stored in the operator buffer. The operator type is set to OP30, and the length is set to 5. TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX25

FUNCTION – To process the GOAL type 25 intermediate text record - Terminate Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXOPUT

DESCRIPTION – The type code is set to OP37. The length is set to 3. The termination code is moved to the operator buffer. TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX26

FUNCTION – To process the GOAL type 26 intermediate text record - Statement Label

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXERR, TXGET, TXPUT, TXOPUT

DESCRIPTION – The indicated statement label is looked up in the label table and its address placed in the operator buffer. The type code is det to OP35, and the length is set to 3. TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX27

FUNCTION – To process the GOAL type 27 intermediate text record - GO TO Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXERR, TXGET, TXOPUT

DESCRIPTION – The indicated statement label is looked up in the label table and its address is placed in the operator buffer. The type code is set to OP16 and the length is set to 3. TXOPUT is called to update the tabular operator file. The 'END' flag is set to terminate processing of the intermediate GOAL data file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX28

FUNCTION – To process the GOAL type 28 intermediate text record - Begin Program Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXOPUT

DESCRIPTION – Type code is set to OP08 and length is set to 2. TXOPUT is called to update tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX29

FUNCTION – To process the GOAL type 29 intermediate text record - End Program Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXOPUT

DESCRIPTION – The type code is set to OP15 and the length is set to 2. TXOPUT is called to update the tabular file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX30

FUNCTION – To process the GOAL type 30 intermediate text record - Activate Table (All) Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXERR, TXLOOK, TXGET, TXOPUT

DESCRIPTION – The type code is set to OP01 and the length is set to 4. The specified table name is looked up and the address of its inhibits list is stored in the operator buffer along with its length. TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX31

FUNCTION – To process the GOAL type 31 intermediate text record - Activate Table (Row) Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXERR, TXLOOK, TXGET, TXOPUT

DESCRIPTION – The type code is set to OP02. The specified table name is looked up and the address of its inhibits list is stored in the operator buffer. For each specified row an entry is generated to contain the address of a variable subscript or a negative constant value. The row count and length are placed in the operator buffer and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX32

FUNCTION – To process the GOAL type 32 intermediate text record - Inhibit Table (All) Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXERR, TXLOOK, TXGET, TXOPUT

DESCRIPTION – The type code is set to OP17, and the length is set to 4. The specified table name is looked up and the address of its inhibits list is stored in the operator buffer along with its length. TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX33

FUNCTION – To process the GOAL type 33 intermediate text record - Inhibit Table (Row) Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXERR, TXLOOK, TXGET, TXOPUT

DESCRIPTION – The type code is set to OP18. The specified table name is looked up and the address of its inhibits is placed in the operator buffer. For each specified row an entry is generated to contain the address of a variable subscript or a negative constant value. The row count and length are placed in the operator buffer and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX34

FUNCTION – To process the GOAL type 34 intermediate text record - Enter/Leave Critical Mode

CALLED BY – TXMAIN

SUBROUTINES CALLED – None

DESCRIPTION – The critical mode control word, C(78), is updated in COMMON block/CONTRL/.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX35

FUNCTION – To process the GOAL type 35 intermediate text record - Perform Subroutine Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXERR, TXNAML, TXNGEN, TXGET, TXOPUT

DESCRIPTION – The subroutine reference is looked up in the External Reference table. The address is placed in the operator buffer. If parameters are required they are generated using TXXGEN and/or TXNGEN. The parameter count, critical flag, and length are entered. TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX36

FUNCTION – To process the GOAL type 36 intermediate text record - Concurrently Perform Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXNAML, TXNGEN, TXOPUT

DESCRIPTION – The type code is set to OP12. If a time internal is required it is generated using TXNGEN. The length is entered and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX37

FUNCTION – To process the GOAL type 37 intermediate text record - Release Concurrent Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXGET, TXOPUT

DESCRIPTION – The type code is set to OP29. Specified labels are looked up in the label table and their addresses entered in the operator buffer. The label count and block length are entered and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX38

FUNCTION – To process the GOAL type 38 intermediate text record - Assign Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXNAML, TXNGEN, TXOPUT

DESCRIPTION – The type code is set to OP06. TXNGEN is used to generate the INTNAM formats. The length is entered and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX39

FUNCTION – To process the GOAL type 39 intermediate text record - Record Present Value of Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXXGEN, TXOPUT

DESCRIPTION – The type code is set to OP27. TXXGEN is called to generate the EXTDES formats. The length is entered and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX40

FUNCTION – To process the GOAL type 40 intermediate text record - Record Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXXGEN, TXNAML, TXNGEN, TXOPUT

DESCRIPTION – The type code is set to OP28. TXXGEN is called to generate the output EXTDES format. A list of INTNAM addresses is generated for each output parmeter. TXNGEN is called to generate the INTNAM formats. 'New line' is indicated by a zero address. The length is entered and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX41

FUNCTION – To process the GOAL type 41 intermediate text record - Apply Present Value of Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXXGEN, TXOPUT

DESCRIPTION – The type code is set to OP03. TXXGEN is used to generate the EXTDES formats. The length is entered and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX42

FUNCTION – To process the GOAL type 42 intermediate text record - Apply Analog (list or table column) Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXXGEN, TXOPUT

DESCRIPTION – The type code is set to OP05. TXNGEN is called to generate the INTNAM format for the data. TXXGEN is called to generate the EXTDES format for the output device. The length is updated and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX43

FUNCTION – To process the GOAL type 43 intermediate text record - Apply Analog (scalars) Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXXGEN, TXOPUT

DESCRIPTION – The type code is set to OP05. TXXGEN is called to generate the output EXTDES format. A list of INTNAM addresses is provided for the output data. TXNGEN is called to generate the INTNAM formats. The length is entered and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX44

FUNCTION – To process the GOAL type 44 intermediate text record - Set Present Value of Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXXGEN, TXNGEN, TXOPUT

DESCRIPTION – The type code is set to OP32. TXXGEN is called to generate the EXTDES formats. The count is entered and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX45

FUNCTION – To process the GOAL type 45 intermediate text record - Set External Designator (list or table column)Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXXGEN, TXNAML, TXNGEN, TXOPUT

DESCRIPTION – The type code is set to OP34. TXXGEN is called to generate the output EXTDES format. TXNGEN is called to generate the INTNAM format. The length is entered and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX46

FUNCTION – To process the GOAL type 46 intermediate text record - Set Discrete (scalars) Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXXGEN, TXNAML, TXNGEN, TXOPUT

DESCRIPTION – The type code is set to OP34. TXXGEN is called to generate the output EXTDES format. A list of INTNAM addresses is provided for the output data. TXNGEN is called to generate the INTNAM formats. The length is entered and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX47

FUNCTION – To process the GOAL type 47 intermediate text record - Read Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXXGEN, TXNAML, TXNGEN, TXOPUT

DESCRIPTION – The type code is set to OP26. TXXGEN is called to generate the input EXTDES format. TXNGEN is called to generate the INTNAM format. The length is entered and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX48

FUNCTION – To process the GOAL type 48 intermediate text record - Average Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXXGEN, TXNAML, TXNGEN, TXOPUT

DESCRIPTION – The type code is set to ØP07. TXXGEN is called to generate the input EXTDES format. TXNGEN is called to generate the INTNAM format. The length and sample count are entered and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX49

FUNCTION – To process the GOAL type 49 intermediate text record - Issue Digital Pattern. (present value of) statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXXGEN, TXOPUT

DESCRIPTION – The type code is set to ØP19. TXXGEN is called to generate the EXTDES formats. The length is entered and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX50

FUNCTION – To process the GOAL type 50 intermediate text record - Issue Digital Pattern (list or table column) statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXXGEN, TXNAML, TXNGEN, TXOPUT

DESCRIPTION – The type code is set to ØP21. TXXGEN is called to generate the output EXTDES format. TXNGEN is called to generate the INTNAM format. The length is entered and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX51

FUNCTION – To process the GOAL type 51 intermediate text record - Issue Digital Pattern (scalars) statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXXGEN, TXNAML, TXNGEN, TXOPUT

DESCRIPTION – TXXGEN is called to generate the output EXTDES format. If no data is specified the type code is set to ØP20. Otherwise, the type code is set to ØP21, and an INTNAM address list is provided. TXNGEN is used to generate the INTNAM formats. The length is entered and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX52

FUNCTION – To process the GOAL type 52 intermediate text record - Time Prefix

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXERR, TXNAML, TXNGEN, TXGETL, TXOPUT

DESCRIPTION – The type code is set to ØP38. TXNGEN is called to generate the INTNAM format for the time value. The length and 'AFTER/WHEN' code are entered and TXOPUT is called to update the procedural operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX53

FUNCTION – To process the GAOL type 53 intermediate text record - Delay Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXXGEN, TXNAML, TXNGEN, TXOPUT

DESCRIPTION – The type code is set to ØP13. If a time delay is specified, TXNGEN is called to generate the INTNAM format. If a condition delay is specified, TXXGEN and TXNGEN are called to generate the required EXTDES and INTNAM(s). The relation code is entered as required. The length is entered and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX54

FUNCTION – To process the GOAL type 54 intermediate text record - Stop Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXERR, TXGET, TXOPUT

DESCRIPTION – The type code is set to ØP36. If restart labels are specified they are looked up in the label table and a list of addresses is provided. The length is entered and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX55

FUNCTION – To process the GOAL type 55 intermediate text record - Request Keyboard Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXERR, TXNAML, TXNGEN, TXGETL, TXOPUT

DESCRIPTION – The type code is set to ØP31. The specified function designator is looked up in the I/O table and its address is entered. TXNGEN is called to generate the INTNAM format for the input data. 'Request' messages are processed in the same manner as the RECORD operator. The length is entered and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX56

FUNCTION – To process the GOAL type 56 intermediate text record - Condition Prefix (If/Then variation)

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXNAML, TXNGEN, TXOPUT

DESCRIPTION – The type code is set to ØP40. TXNGEN is called to generate the INTNAM formats for the required comparands. The length and relation code are entered and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX57

FUNCTION – To process the GOAL type 57 intermediate text record - Condition Prefix (Verify)

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXXGEN, TXNAML, TXNGEN, TXOPUT

DESCRIPTION – Set type code to ØP39. Call TXNGEN to generate INTNAM format for time limit if required. Call TXXGEN to generate EXTDES format for sensor type external designator. Call TXNGEN to generate INTNAM formats for comparands as required. Enter 'ELSE/THEN' code, relationship code, and length. Call TXOPUT to update tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX58

FUNCTION – To process the GOAL type 58 intermediate text record - Output Exceptions

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXXGEN, TXNAML, TXNGEN, TXOPUT

DESCRIPTION – Type code is set to ØP23. TXXGEN is called to generate EXTDES format for output device. TXNGEN is called to generate INTNAM format for exception message if one is specified. Otherwise, a zero address is generated to indicate default. The length is entered and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX59

FUNCTION – To process the GOAL type 59 intermediate text record - Perform Program Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXERR, TXGET, TXOPUT

DESCRIPTION – Type code is set to ØP24. The specified program is looked up in the EXTERNAL REFERENCE table and the address is entered in the operator buffer. The length is set to 3 and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX60

FUNCTION – To process the GOAL type 60 intermediate text record - Let Equal Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXNAML, TXNGEN, TXOPUT

DESCRIPTION – The type code is set to ØP22. A code/address list is generated to describe the arithmetical expression. TXNGEN is used to generate the INTNAM formats for variables and constants. The list count and length are entered and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX61

FUNCTION – To process the GOAL type 61 intermediate text record - Begin Subroutine Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXLOOK, TXERR, TXOPUT

DESCRIPTION – The type code is set to ØP09. A list is provided which contains a pair of entries for each subroutine parameter. The 1st word of the pair is 0/1 for Value/Function Designator. The second word of the pair is the addres of 1) INTNAM DDCB or 2) ENTRY in the Function Designator I/O table. The list count and length are entered and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX64

FUNCTION – To process the GOAL type 64 intermediate text record - Disable Interrupt Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXGET, TXERR, TXOPUT

DESCRIPTION – The type code is set to ØP14. If statement labels are specified, they are looked up in the label table and an address list is provided. The list count and length are entered and TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX65

FUNCTION – To process the GOAL type 65 intermediate text record - When Interrupt Statement

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXGET, TXERR, TXOPUT

DESCRIPTION – The type code is set to ØP41 and the length is set to 3. The specified function designator is looked up in the I/O table and its address is entered. TXOPUT is called to update the tabular operator file.

GOAL INTERPRETIVE CODE TRANSLATOR ROUTINE

NAME – TX68

FUNCTION – To process the GOAL type 68 intermediate text record - Return To

CALLED BY – TXMAIN

SUBROUTINES CALLED – TXGET, TXERR, TXOPUT

DESCRIPTION – The type code is set to ØP42 and the length is set to 3. If a label is specified it is looked up in the label table and its address is entered. TXOPUT is called to update the tabular operator file.

APPENDIX A

GOAL INTERPRETIVE CODE GUIDELINE

A. INTRODUCTION

The purpose of this document is to present the rationale for interpretive execution of GOAL programs and to identify and describe relevant implementation options. This discussion is intended to assist in the development of checkout systems using output from the GOAL Interpretive Translator in a real time test applications environment.

The question, Interpret vs. Compile, is discussed in Section A.1. This discussion establishes the groundwork for Section A.2 which covers real time user interactions. The impact of processor characteristics is discussed in Section A.3. Section A.4 includes discussion of a variety of implementation related topics.

For specific details regarding the GOAL Interpretive Translator refer to the document, GOAL INTERPRETIVE CODE TRANSLATOR DESIGN SPECIFICATION.

A.1 COMPILE VS. INTERPRET

With a pure interpretive approach 'raw' source codes are processed at execution time. With a pure compiler approach all source codes are translated to executable codes by the compiler. The interpretive approach provides the greatest ease for on-line development and modification of programs, however, execution efficiency is generally less than compiled codes. The compiler approach provides the greatest execution efficiency, but development and/or modification of a program may require many compiler 'turnarounds'. In the interpretive approach errors may not be detected until the codes in which they occur are executed. The compiler has the advantage of preliminary diagnostics for a wide class of statement errors. In order to obtain acceptable execution efficiency in the interpretive mode it may be necessary to place constraints on the design of the language.

The previous comments apply to a pure interpretive or compiler approach. It is possible to utilize a composite approach where the source codes are compiled to generate an intermediate form which is executed interpretively. By a proper choice of the intermediate form, the significant advantages of both the compiler and interpretive approaches may be obtained. Ideally this form would be selected by comparing the weighted advantages and disadvantages of the candidate forms. However, the weighting factors are dependent upon implementation equipment and applications. Thus the following approach is used: A highly generalized format is provided for the interpretive codes. Compiler/Translator options are provided to enable the user to select output that is compatible with his implementation. This guideline document is provided to assist the user in evaluation of the impact of his selections.

A.2 REAL TIME USER INTERACTIONS

This section is concerned with real time interactions between the test operator and the checkout system. A list of candidate operations is developed and special data requirements related to the use of GOAL are discussed.

A principal mode of interaction between the test operator and the real time checkout system is conversational input/output via on-line (display) terminal. Dedicated switches, indicators, function keys, etc., may also be used. However, their purpose may generally be accomplished using keyed in requests and displayed messages.

The operator interactions with the checkout system discussed in this section are classified according to the following three levels:

1. Operating System Level - this includes those operations normally used in test operations, but which are not specifically related to or dependent upon the use of programs written in a high level test language (GOAL).

2. Language Executive Level - this includes those operations which are related to the use of a high level test language, (GOAL, but are not associated with specific checkout programs.

3. Test Program Level - this includes those operations associated with the execution of a specific test program. The form of these interactions is determined by the logic of the program.

Operator interactions at each of these three levels are discussed in the following sections. Emphasis is placed on those cases where data provided by the Interpretive Translator may be used to support the operation. The Translator output options which provide this data are identified and the use of the data is described.

A.2.1 Operating System Level

At a minimum the test operator may be required to perform the following operations to support the execution of a GOAL test program at the operating system level.

- Initiate program execution
- Interrogate program status
- Verify program completion
- Force program termination
- Note exceptional system conditions

The above operations are common to the execution of any real time program and are not directly related to the user options of the Interpretive Code Translator.

A.2.2 Test Language Executive Level

Test operator interactions at this level include:

- o Unrestricted stop and restart
- o Internal data status and modification
- o Measurement status and command
- o Logic and address modifications

It may be feasible to implement these capabilities using absolute address data provided by standard Translator output listings. However, it may prove advantageous to utilize optional output tables to support those operations in a more direct and convenient way.

A.2.2.1 Asynchronous Stop and Restart. This particular type of program stop occurs as a result of operator request. The operator may wish to determine the location of the current statement being executed, and/or transfer control to another part of the program. The location of labeled statements is always available for this purpose. However, if it is desired to transfer control to an unlabeled statement, the optional Internal Statement Number table should be included in the Translator output.

A.2.2.2 Internal Data Status and Modification. The test operator may wish to interrogate the status of internal data items or to modify their values. In order to reference internal data by symbolic name, the optional internal names table should be included in the Translator output. This also enables the generation of column titles and row names for output of a complete table.

A.2.2.3 Measurement Status and Command. The test operator may wish to interrogate the status of measurements or issue commands to the system under test. In order to reference these I/O points using the same symbolic names as in a GOAL program, the optional Function Designator Names table should be included in the Translator output. If the internal names table is also included, the '(TABLE) FUNCTIONS' reference can be supported for on-line usage.

A.2.2.4 Logic and Address Modification. In exceptional circumstances it may be necessary for the test operator to modify the logic or addressing used in a GOAL program. Logic modifications may include the following:

- o Delete specified statement(s).
- o Change Internal Name Attributes.
- o Insert branch points and stops.

Address modifications may be included:

- o Change internal data reference address in procedural operators.
- o Modify hardware addresses associated with function designators.

Inclusion of all optional Translator tables will facilitate these types of operations. Note that all references to a given function designator are routed through a unique address word. This facilitates the hardware address reconfiguration procedure.

A.2.3 Test Program Level

Execution of the following Interpretive operators may involve some form of user activity or response:

- o PRINT/DISPLAY/RECORD
- o OUTPUT EXCEPTIONS
- o REQUEST KEYBOARD
- o STOP AND INDICATE RESTART LABELS

If the system implementation does not support these statements, they may be flagged as invalid during the translation process according to the standard Translator program user options. If these statements are to be supported, the Translator provides optional outputs which may be used in their execution.

A.2.3.1 PRINT/DISPLAY/RECORD. This GOAL statement is supported by two operators:

- o OP27 - Record Present Value
- o OP28 - Record Data

Operator No. 27 is used to print, display, or record the present value of a measurement to a specified on-line device. The measurement type and address are always available. However, if its symbolic name, as used in the GOAL program, is desired. The optional function designator names table must be included in the Translator output.

Operator No. 28 is used to print, display, or record the current values of internal data variables or fixed data. In all cases, the data type, length and address are available. However, if the symbolic name, row, or column identifiers are desired, the optional Internal names table must be included included in the Translator output.

A.2.3.2 OUTPUT EXCEPTIONS. Operator No. 23 is used to support this GOAL statement phrase. The text of the error messages is always available. However, if the name of the Text Data, List, or table/row/column is desired, the optional internal names table must be included in the translator output.

A.2.3.3 REQUEST KEYBOARD. Operator No. 31 is used to support this statement. The output part of this statement presents a case similar to the PRINT/DISPLAY/RECORD statement. In addition, the input part of the statement presents certain unique aspects. If it is desired that the symbolic names of the input variables be available as part of the 'REQUEST' message, the optional internal names table must be included in the translator output. This would be desirable if the REQUEST statement being processed did not use any output messages.

A.2.3.4 STOP AND INDICATE RESTART LABELS. Operator No. 36 is used to support this statement. The label numbers associated with specified restart labels are always available for inclusion in the output message. Also any valid label included in the GOAL program may be identified for unrestricted restart. However, if unrestricted restart capability is desired for unlabeled GOAL statements, the optional Internal Statement Number table must be included in the Translator output.

A.3 PROCESSOR CHARACTERISTICS

The processor characteristics discussed in this section are considered to be optional or variable in the extent to which they are available or utilized. The actual selection of these options will depend upon implementation requirements. However, in this section, each of the options is described and impacts related to the format of the interpretive code are discussed. The options include:

- Storage size (core)
- Execution speed
- Instruction set
- Data representations (floating point, character, etc.)
- Direct access storage (disk)
- Standard peripheral equipment
- Operating System
- Program Libraries

A.3.1 Storage Size

If sufficient CPU storage is available the entire GOAL program may be loaded as a series of contiguous blocks. This represents an ideal situation and probably cannot be expected as the general case. It is more likely that only the 'resident' portion of the program will be loaded into a dedicated area and the remainder, (procedural operators), will be loaded in blocks for execution as a series of overlays. Multiple buffering and/or paging techniques may be used to support this mode of execution. The choice of optimum block size depends on many application/implementation related factors. The Interpretive Code Translator provides a user control option which enables the specification of the output data block (buffer) size.

A.3.2 Execution Speed

This factor is related to the previously discussed storage size considerations. The time required to load a block of interpretive code should be comparable to the time required for its execution if the overlay technique is used. In a double buffer situation this would generally result in the next sequential block being available when required.

A.3.3 Instruction Set

The GOAL 'LET EQUAL' statement allows addition, subtraction, multiplication, division, and exponentiation. If extensive use is made of these features, the multiply and divide instructions would probably be desirable.

A.3.4 Data Representation

The Interpretive Translator provides output that may be directly loaded into CPU's having word sizes of 16, 24, and 32. Special loaders or preprocessors are required to convert this data for use in machines with other word sizes. The Interpretive Code Translator also allows the user to specify a character size, (within a word) of 6, 7, or 8. The number of characters per word (left justified) may be specified as 1, 2, 3 or 4. The specified word size, character size, and characters per word must be mutually consistent. The standard character set provided by the Translator is ASCII. The Translator provides an option to provide a different character set, if desired.

All numeric values (variables) are represented in a Mantissa/Exponent format. The Mantissa part contains a two's complement integer one word in length. The Exponent part contains a two's complement integer whose value indicates the binary shift factor associated with the Mantissa, (+/right, -/left). The Translator provides an option which provides a two word mantissa, (double precision), if desired. Note: Only one word of precision is provided for initial values.

A.3.5 Direct Access Storage

The availability of direct access storage, (disk, drum, etc.), has the following significant impacts on the use of the interpretive data:

- o Slew time (from block to block) is reduced and predictable in the overlay execution mode.
- o Multitask execution is facilitated by the use of paging or overlay techniques.
- o A substantial portion of the optional 'resident area' tables may be stored on direct access devices, thus reducing dedicated core requirements.

A.3.6 Standard Peripheral Equipment

The Translator data representation and character set options should accommodate the requirements of most unit record equipment. The standard media for interpretive data output is seven or nine track tape. The target implementation system should have some means of loading this tape, or processing it into loadable form.

A.3.7 Real Time Executive

Use of the Interpretive Data implies that a suitable real time executive is available to interpret the operators and execute their functions. The interpretive operator formats have been selected to minimize the processing requirements for this executive program. If a given operator(s) is not to be supported by the executive, a Translator option is provided to flag its occurrence and suppress the generation of interpretive data. In addition, the assigned operator type codes may be modified by simple changes to the translator.

A.3.8 Program Libraries

The GOAL 'PERFORM' statement enables reference to a program which is not necessarily included in the interpretive data. These references may be identified using the External Reference Control Block which is always provided in the interpretive output data. The real time executive (or loader) must identify and retrieve these programs so that they may be available when required.

A.4 RELATED TOPICS

A.4.1 Output Listings

The GOAL Interpretive Code Translator Output Listings have been designed to provide the user with complete information regarding the content and format of the interpretive data file. These listings will enable the user to follow the execution of his program in real time and/or modify any of its parameters via suitable on-line executive facilities. If the appropriate symbolic tables have been included in the interpretive output, these modifications may be accomplished via symbolic name. Otherwise, they must be specified using relative word addresses obtained from the Translator Output Listings.

A.4.2 On-Line Data Bank

If the optional Function Designator Names table is selected by the user, the Interpretive Translator output data will contain the symbolic names of all measurements referenced in a GOAL program. This provides, in effect, a 'Data Bank' which is unique to the program under consideration. Since a real time checkout application may require many GOAL programs it seems reasonable that the on-line Data Bank contain the summation of the references contained in the individual programs. However, its contents are not limited to this data, and it may contain any other information that is useful in support of the real time checkout application. This approach requires that all measurements referenced in GOAL programs have unique names and that these do not conflict with other information contained in the on-line Data Bank.